Dublin City University

School of Electronic Engineering

Crowd Behaviour and Congestion Analysis Through Deep Machine Learning

Mark Marsden

December 2018

This dissertation is submitted for the degree of

Doctor of Philosophy

Supervised by Prof. Noel E. O'Connor

and Prof. Kevin McGuinness

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work,

Signed : ..... ID: 59352341

Date : .....

Acknowledgements

I would like express my deepest appreciation for all the guidance and support I have received during the past four years from my co-supervisors Noel O'Connor and Kevin McGuinness. Both have shown never ending encouragement, patience and kindness throughout my studies, making it an incredibly enjoyable and rewarding experience from start to finish. It was always fun and exciting to do research in this group. Kevin's attempts to convince me that metal is great and that all other genres of music are terrible did not go unnoticed, but ultimately failed. I'd also like to point out that Dublin have won the all Ireland every year I've studied under Noel, I'm clearly a good luck charm. I would also like to thank everyone at the Insight Centre that I have worked with over the past few years. You have all been wonderfully helpful and considerate throughout my PhD journey. I've spent 9 years at DCU from undergraduate right through to PhD and I owe an awful lot to this young but rapidly growing university. I wish it every success in the future. No matter where I go I'll always be shaped by my experience at DCU and the great people I met.

On a personal note, I would not have made it this far without the endless encouragement and support of my parents, Maeve and Robbie, and my grandparents Kay, Thomas and Anna. From day one you were all completely behind my decision to pursue academia when other options were available and for this I am eternally grateful. Hopefully this thesis can help answer the question "What does Mark actually do in there?".

Finally the most significant dedication must go to my wonderful girlfriend Ciara, who's daily support and loving patience (while committed to her own academic journey) have been vital to any achievement, large or small, made over the past 7 years together. From settling my nerves before a big presentation to proof reading piles of computer science gibberish, you contributed so much to this PhD. I love you so much and I have no idea where I'd be without you.

List of Publications

Peer Reviewed Publications

- **M. Marsden,** K. McGuinness, S. Little, and N.E. O'Connor, "Holistic features for real-time crowd behaviour anomaly detection". in 2016 IEEE International Conference on Image Processing (ICIP), 2016, pp. 918–922.

- **M. Marsden,** K. McGuinness, S. Little, and N.E. O'Connor, "Fully convolutional crowd counting on highly congested scenes".The 12th International Conference on Computer Vision Theory and Applications (VISAPP), 2017.

- **M. Marsden**, K. McGuinness, S. Little, and N.E. O'Connor, "ResnetCrowd: A Residual Deep Learning Architecture for Crowd Counting, Violent Behaviour Detection and Crowd Density Level Classification". The IEEE conference on Advanced Video and Signal-based Surveillance (AVSS), 2017.

- C. Ballas, **M. Marsden**, D. Zhang, N.E. O'Connor and S. Little, "Performance of Video Processing at the Edge for Crowd-Monitoring Applications". The 4th IEEE World Forum on Internet of Things, 2018.

- **M. Marsden**, K. McGuinness, S. Little, C.E. Keogh and N.E. O'Connor, "People, Penguins and Petri Dishes: Adapting Object Counting Models To New Visual Domains And Object Types Without Forgetting". Computer Vision and Pattern Recognition (CVPR), 2018.

Workshop Papers

- K. McGuinness, E. Mohedano, A. Salvador, Z. Zhang, **M. Marsden**, P. Wang, I. Jargalsaikhan, J. Antony, X. Giro-i-Nieto, S. Satoh, N.E. O'Connor and A. Smeaton. "Insight DCU at TRECVID 2015". In TRECVID 2015 Overview Papers and Slides, 2015, pp. 1-16.

- **M. Marsden**, E. Mohedano, K. McGuinness, A. Calafell, X. Giró-i-Nieto, N. E. O'Connor, J. Zhou, L. Azevedo, T. Daudert, B. Davis, M. Hürlimann, H. Afli, J. Du, D. Ganguly, W. Li, A. Way, A. F. Smeaton. "Dublin City University and Partners Participation in the INS and VTT Tracks at TRECVid 2016", In TRECVID 2016 Overview Papers and Slides, 2016.

Acronyms and Abbreviations

ACC	Accuracy
CNN	Convolutional Neural Network
LSTM	Long short-term memory unit
FCN	Fully Convolutional Network
GPU	Graphics Processing Unit
CCTV	Closed Circuit Television
RNN	Recurrent Neural Network
SIFT	Scale-Invariant Feature Transform
FPS	Frames Per Second
STV	Spatio-Temporal Volume
MSE	Mean Square Error
MAE	Mean Absolute Error
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
GMM	Gaussian Mixture Model
HOG	Histogram of Oriented Gradients
CPU	Central Processing Unit
EER	Equal Error Rate
SVM	Support Vector Machine

RGB The red/green/blue colour space

ReLU Rectified linear unit

API Application Programming Interface

TPR True Positive Rate

FPR False Positive Rate

DLE Density Level Estimation

SOTA State-Of-The-Art

MTL Multi Task Learning

DA Domain Adaptation

TL Transfer Learning

IOT Internet Of Things

MAP Mean Average Precision

Contents

List of Figures

List of Tables

Abstract

Crowd Behaviour and Congestion Analysis Through Deep Machine Learning

Mark Marsden

This thesis looks to advance understanding in the field of computer vision based crowd analysis through a combination of deep learning techniques, multi-task learning, and domain adaptation. Issues that have limited progress in this field to date include visual occlusion, scale and perspective issues, variation in scene content as well as a lack of labelled training data. Another negative trend that has emerged in this field as well as in computer vision in general is the development of bespoke, single-task techniques that cannot be easily extended or re-used.

The core contributions of this work are as follows. First, deep learning methods are developed for several crowd analysis tasks including crowd counting, crowd density level estimation, crowd behaviour recognition and crowd behaviour anomaly detection. The proposed data-driven methods are shown to be superior to techniques which rely on hand-crafted features, overcoming many of the observed challenges and achieving state-of-the-art results. Second, multi-task learning strategies are applied to crowd behaviour and congestion analysis tasks, increasing the overall predictive performance and removing redundant model parameters. Finally, domain adaptation techniques are investigated as a means to extend a given crowd analysis model to perform the same task in new visual domains (e.g. medical, wildlife) and vice-versa, with original domain performance preserved.

“Invention, my dear friends, is 93% perspiration, 6% electricity, 4% evaporation and 2% butterscotch ripple”

Gene Wilder, Willy Wonka and the Chocolate Factory

Chapter 1

Introduction

The objective of this chapter is to provide a high level introduction to the subject of the thesis and to describe the specific research objectives and core motivations. The next section introduces the research area of computer vision as well as the topic of vision-based crowd analysis, which itself is composed of several related analysis tasks. Following this, the main motivations behind work in this area are discussed, some potential applications are presented as well as the core research objectives and proposed hypotheses of the thesis. Finally, the overall structure of the thesis is outlined.

1.1 Overview

Computer vision is a broad research area that deals with the extraction of high level understanding from digital images and video. Research in this area covers a range of analysis tasks including object recognition, semantic segmentation, 3-D pose estimation and VQA (visual question answering) among others. Computer vision techniques have been utilised in a range of application domains including medical imaging, autonomous vehicles, facial recognition, content-based information retrieval and crowd analysis. The origins of this field date back to

the 1960s when Marvin Minsky, a co-founder of the Massachusetts Institute of Technology AI lab, proposed *The Summer Vision Project* [1], a short term project that looked to developed accurate image segmentation and object identification algorithms over the course of a single summer. Work on these tasks continues to this day across the world. This thesis focuses on the development of computer vision techniques to better understand the nature of large and dynamic crowds.

Vision-based crowd analysis looks to extract global scene attributes from images containing large groups of people, such as those shown in figure 1.1. Examples of such attributes include the overall number of people present in the scene as well the collective behaviour of the crowd.

Figure 1.1: Examples of the significant variation observed in images of large crowds

Vision-based crowd analysis can be separated into a number of related analysis tasks which can be defined as follows:

- **Crowd Counting**: Estimate the true number of people present in an image of a crowded

[1] https://dspace.mit.edu/bitstream/handle/1721.1/6125/AIM-100.pdf

scene, expressed as an integer value.

- **Crowd Density Level Estimation**: Classify the congestion level observed in an image of a crowded scene, expressed on a discrete scale (0-N).

- **Crowd Behaviour Recognition**: Categorise the collective behaviour observed in an image or video of a crowded scene, expressed as a set of likelihood scores for a range of behaviour concepts.

- **Crowd Behaviour Anomaly Detection**: Detect collective crowd behaviour in an image or video that strays from an established norm, expressed as the likelihood that the given frame or sequence of frames contains abnormal behaviour.

These 4 crowd analysis tasks can be split into two classes, crowd congestion analysis and crowd behaviour analysis, as shown in the taxonomy presented in figure 1.2. While there is an obvious relationship between the congestion level and behaviour of a large crowd, these factors will be treated separately for the purposes of this study. can be related Related computer vision tasks that do not fall under crowd analysis include person re-identification and human action recognition. These tasks will not be investigated as part of this thesis as the focus of these tasks is on an individual subject rather than the collective crowd.

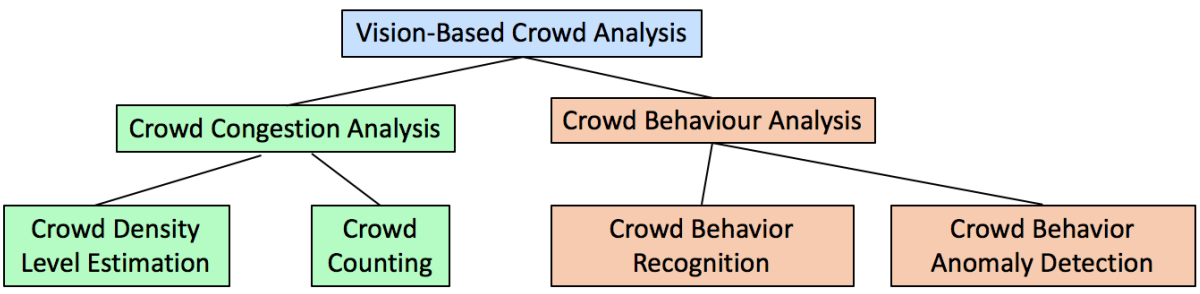

Figure 1.2: Taxonomy of the various vision-based crowd analysis tasks

1.2 Motivation

According to a 2014 report by the U.N., 54% of the world's population currently live in urban areas with this projected to increase to 66% by 2050[2]. This corresponds to a growth in global urban population of 2.5 billion people over the next 32 years. With this rapid increase in urban population, highly congested crowds will become a significant part of daily life, presenting enormous challenges to the maintenance of public safety and the efficient movement of people in modern cities. Every year dozens of people are injured or killed in densely populated urban areas due to stampedes and crushes[3]. A recent example of this was the 2014 New Year's Eve stampede in Shanghai, China where 36 people tragically died. This loss of life could potentially be prevented with better analysis and understanding of crowd behaviour and congestion levels across large metropolitan areas.

An unprecedented global rise in CCTV (Closed-Circuit Television) surveillance has accompanied this rapid growth in urban population. In cities such as London there have been extensive networks of CCTV cameras installed[4]. This rapid deployment of infrastructure has resulted in more CCTV footage being produced than can be analysed by human observers. Therefore the automated analysis of these ever growing archives has become a necessity in order to fully benefit from this large-scale deployment. This increase in CCTV data generation, however, pales in comparison to the quantity of video now produced and shared online by individual users from their smartphones and personal devices. It is projected that mobile video traffic will reach 38.1 Exabytes per month by 2021, up from 4.4 Exabytes per month in 2016 [5]. One Exabytes is equivalent to one billion gigabytes. A significant number of video clips shared on platforms

[2]http://www.un.org/en/development/desa/news/population/world-urbanization-prospects-2014.html

[3]https://en.wikipedia.org/wiki/List_of_human_stampedes

[4]https://www.caughtoncamera.net/news/how-many-cctv-cameras-in-london/

[5]https://www.cisco.com/assets/sol/sp/vni/forecast_highlights_mobile/

such as Facebook contain violent and potentially upsetting behaviour such as large fights and anti-social behaviour. Monitoring for this footage has proven a significant challenge, with automated solutions not yet able to accurately detect the undesired content.

Clearly there is a need for accurate and scalable crowd video analysis systems both in the CCTV surveillance and online video domains. Developing such systems will allow us to produce safer and more efficient cities and prevent the proliferation of undesirable and hateful video content online.

1.3 Applications

The development of highly accurate vision-based crowd analysis systems has the potential for significant societal impact across several application areas. These potential applications are discussed in this section.

1.3.1 Internet-of-Things and Smart Cities

Figure 1.3: The IoT-based Smart City concept visualised (Shah, 2018)

The term *Internet-of-Things* (IoT) refers to the vast network of Internet connected physi-

cal devices found across the globe, all embedded with computing hardware and individually identifiable within the network. This label was coined by Kevin Ashton of Procter & Gamble in 1999[6]. This network of devices ranges from household appliances and CCTV cameras to biochip transponders in farm animals. Utilising this vast network of devices can allow more direct integration between the physical and digital worlds, leading to significant societal and economic impact. It is estimated that by 2020 there will be 30 billion IoT devices in the wild[7] and that the global market value of the IoT sector will reach 7.1 trillion US dollars (Hsu and Lin, 2016).

A major component of the overall IoT sector is the area of *Smart City* technologies. This refers to an urban area that utilises data gathered by IoT devices to manage resources and assets efficiently. This includes the movement of people, the consumption of energy and the safety of individual citizens. This concept of a connected, data-driven Smart City is illustrated in figure 1.3. IP-enabled CCTV cameras represent a significant portion of the overall IoT network in a Smart City, providing significant amounts of video data. Accurate analysis of this video data can contribute significantly to the overall success of a Smart City concept, particularly when it comes to analysing the movement of people, the management of large public spaces and the detection of violent or dangerous crowd behaviour.

1.3.2 Online Video Monitoring

The introduction of video live-streaming on platforms such as Facebook has lead to an unmanageable quantity of video footage being produced and shared. These has enabled footage of violent and hateful scenes, sometimes involving large crowd fights, murders, and rapes finding

[6]http://www.rfidjournal.com/articles/view?4986

[7]https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated

Figure 1.4: The live-streaming of video from mobile devices is now an everyday occurrence.

its way to everyday users of these services. Automated solutions are not yet sufficiently accurate or computationally efficient, leading to large scale deployment of human agents to tackle the problem [8]. The development of highly accurate and robust crowd analysis algorithms that can efficiently detect violent behaviour can contribute to the mammoth task of monitoring the millions of hours of footage shared online on a daily basis.

1.4 Hypotheses and Research Questions

A significant body of literature now exists for the topic of vision-based crowd analysis with many techniques developed for each of the crowd analysis tasks discussed in section 1.1. Despite this wealth of research activity, significant challenges still exist that limit the accuracy and reliability of current crowd analysis algorithms. These challenges include visual occlusion, scale and perspective issues caused by the camera position, as well as high levels of variation in scene content and illumination levels. The recent re-adoption of artificial neural networks by the machine learning community (Krizhevsky et al., 2012), enabled largely by the use of hardware

[8] http://abcnews.go.com/Technology/facebook-hire-3000-workers-monitor-content/story?id=47178969

accelerated numerical optimisation, has led to noticeable performance increases across many computer vision tasks including some promising initial work in crowd analysis. These data-driven approaches, now commonly referred to as *deep learning*, allow for analysis pipelines to optimise their parameters specifically for a given task. The optimal application of these techniques to tasks other than conventional image classification, however, remains an open problem for the computer vision research community. Another prevalent issue in vision-based crowd analysis is the development of bespoke, single problem solutions, which require a significant amount of engineering work and cannot be easily re-used or extended to perform additional analysis tasks in other visual domains (e.g. medical, scientific). Given these opportunities and challenges observed within the vision-based crowd analysis space, the hypotheses of this work can be stated as follows.

Hypotheses

1. Data-driven models such as convolutional neural networks (CNN) are superior to hand-crafted methods for vision-based crowd analysis tasks both in terms of predictive performance and adaptability to various problem types.

2. Multi-task learning (MTL) techniques can be used to improve the predictive performance of vision-based crowd analysis models and reduce the overall trainable parameter count across related crowd analysis tasks.

3. Domain adaptation (DA) techniques can be used to extend a crowd analysis model to other visual domains and vice versa while retaining model accuracy for all domains and significantly reducing the overall parameter count.

To address this set of hypotheses the following research questions will be investigated experimentally in chapters 3-6:

Research Questions

1. **To what degree can the application of deep neural networks improve the accuracy and robustness of computer vision based crowd analysis over methods that utilise hand-crafted features and what are the best practices when using this technique?**

   This research question explores the effectiveness of deep neural networks as an approach to solve various crowd analysis tasks. Deep learning implementations are investigated for the tasks of crowd behaviour recognition (chapter 3), crowd behaviour anomaly detection (chapter 3), crowd density level estimation (chapter 4), and crowd counting (chapter 4). Comparisons are then made for each task to methods that rely on hand-crafted features as well as the leading deep learning approaches from the literature.

   An important caveat to remember when comparing deep learning and hand crafted methods is the requirement for labelled samples when training a deep learning model. This labelled data may not always be readily available. However, with advances in transfer learning and generative models this problem is becoming less restrictive. Another issue to note is that deep learning models are not designed to replace well-established physical and physiological models for natural phenomena, but rather these deep models can be used to recognise patterns previously impossible to detect.

2. **Can multi-task learning techniques be used to improve the predictive performance of crowd analysis models and what are the associated benefits and tradeoffs?**

   A major limitation in vision-based crowd analysis is the lack of commonality and reusability between developed models. This research question investigates how multi-task learning techniques can improve the predictive performance of crowd analysis models and reduce the overall number of trainable parameters required (chapter 5). Auxiliary

loss terms are investigated in a single task setting before the joint training of multiple related crowd analysis tasks in a single model is evaluated.

3. **Can domain adaptation techniques be used to adapt computer vision models trained in other visual domains to accurately perform crowd analysis tasks and vice versa? What are the associated benefits and tradeoffs?**

   Another issue associated with recent crowd analysis algorithms is that the developed models are optimised to work within a given visual domain (e.g. CCTV footage) and cannot be used to perform similar computer vision tasks in other visual domains such as medical imaging. This research question investigates the use of recently developed domain adaptation strategies as a means to adapt a crowd analysis model to other visual domains and vice versa (chapter 6). These methods are compared to more traditional transfer learning techniques (fine-tuning, feature extraction). This thesis will focus on domain adaptation in context object counting which has not yet been investigated by the research community.

1.5 Thesis Structure

The remainder of the thesis is structured as follows:

- **Chapter 2** : A comprehensive literature review is carried out. First, background theory relevant to the research carried out in this thesis is presented. Second, the dominant trends and leading approaches used for each of the four vision-based crowd analysis tasks highlighted in section 1.1 tasks are discussed. Third, research within the area of various multi-task learning and domain adaptation is presented. Finally, the commonly used datasets in the field of vision-based crowd analysis are discussed before a final subset is decided upon for the experimental phase of the thesis.

- **Chapter 3**: Deep learning based crowd behaviour analysis is investigated. Various convolutional neural network configurations, preprocesing steps and and implementation strategies are evaluated for each task in an attempt optimise performance and find the dominant trends and best practices for crowd behaviour analysis. A baseline run using hand crafted features is included for each task to compare performance with deep learning methods. Following the development of a refined method for each task using a validation set, comparisons are made with the leading techniques from the literature using a larger test set.

- **Chapter 4**: Deep learning based crowd congestion analysis is investigated. Various convolutional neural network configurations, preprocesing steps and and implementation strategies are again evaluated for each task in an attempt to optimise performance and find the dominant trends and best practices for crowd congestion analysis. Comparisons are then made with a hand-crafted baseline as well as the learning techniques from the literature.

- **Chapter 5**: Multi-task learning is investigated within the context of crowd analysis. Auxiliary loss terms are evaluated within the context of single-task learning before the joint training of related crowd analysis tasks within a shared model is performed. Multi-objective models are compared to single-objective baselines both in terms of predictive performance achieved and the overall number of model parameters required.

- **Chapter 6**: Domain adaptation is investigated as a means to extend a crowd analysis model to other visual domains and vice versa. Recently proposed domain adaptation techniques are compared to more traditional transfer learning methods (feature extraction, fine-tuning).

- **Chapter 7**: In this chapter a summary of the thesis is presented. Each of the hypotheses

and related research questions proposed in chapter 1 are addressed with respect to the experimental results produced in chapters 3-6. The core research contributions of the thesis are then presented before a discussion on the possible future research is finally presented.

Chapter 2

Literature Review

This chapter firstly presents background theory relevant to the work carried out in this thesis (section 2.1), including artificial neural networks, gradient descent and optical flow estimation. This is followed by a review of the leading works in the areas of crowd behaviour analysis (Section 2.2), crowd congestion analysis (Section 2.3), multi-task learning (Section 2.4) and domain adaptation (Section 2.5). The main datasets used for performance benchmarking in the area of vision-based crowd analysis are also discussed (Section 2.6) before a subset is chosen for the experimental phase of the thesis.

2.1 Background Theory

This section presents an overview of the core background theory upon which the research presented in this thesis relies.

2.1.1 Artificial Neural Networks

An artificial neural network (ANN) is a biologically inspired computing model optimised to perform a given analysis task in a data-driven fashion rather than with hand engineered rules. A

given ANN consists of a set of connected units or nodes referred to as *neurons*. Early work in this field developed single neuron models called perceptrons before combining multiple neurons to form full networks.

The Single Neuron Model

In this type of model a single neuron (alternatively referred to as a perceptron) applies a set of learned weights w via dot product to a fixed length input vector x before adding a learned bias b as described in equation 2.1. This fixed input vector x is some existing data representation (e.g. meteorological sensor data) for which a prediction needs to be made (e.g. will it rain or not). The output of this single neuron is then passed through an activation function f, given in equation 2.2, which within this biological abstraction decides whether this neuron *fires* or not based on the input x.

$$u = w \cdot x + b \tag{2.1}$$

$$y = f(w \cdot x + b) \tag{2.2}$$

Various non-linear activation functions can be employed to model non-linear relationships including sigmoid, hyperbolic tan and rectified linear unit (ReLU) functions. On the other hand, a linear activation function ($f(u) = u$) can be applied if the underlying relationship can be modeled as linear. The full perceptron model is visualised in figure 2.1. The parameters of a single neuron model (w, b) can be optimised to perform either regression or binary classification by employing a set of training samples $\{x_i, y_i\}_{i=0}^{N}$ to minimise an objective function by adjusting the weights and bias term. An objective function typically contains a loss term which measures the performance of the model and a regularisation term which adds stability to the optimisation.

Binary classification requires a decision threshold to be applied to the neuron output y which introduces another model selection parameter to tune.

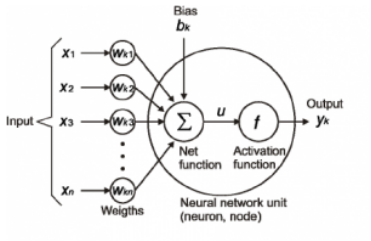

Figure 2.1: A single neuron model

Multi-class classification can be performed by extending the model to contain several neurons which are optimised simultaneously, with each neuron producing a likelihood score for a given class/concept. Both the input x and output y of this model are vectors in this setting. For multi-class classification an activation function such as *softmax*, given in equation 2.3, is typically applied to the output of all neurons simultaneously. This activation function normalises the output to form a categorical probability distribution over the possible classes with an overall sum of 1.0.

$$f(u_i) = \frac{e^{u_i}}{\sum_{j=0}^{N} e^{u_j}} \tag{2.3}$$

Multi-Layer Networks

The single neuron models described above can be used to map simple relationships but more complex functions may prove more difficult to approximate. To address this issue additional model complexity can be created by feeding the activations of a multi-neuron model into another multi-neuron model forming a multi-layered model that can be jointly optimised. This multi-layered model is then referred to as a neural network or fully connected network. Such a model, visualised in figure 2.2, consists of an input layer of neurons, an output layer of neurons and an arbitrary number of intermediary *hidden* layers. These hidden layers produce progressively

more abstract internal representations of the input vector x which can allow for challenging non-linear relationships to be modeled. The number of hidden layers as well as the number of neurons per hidden layer can be varied to include additional trainable parameters and model more complex functions. The quantity of trainable parameters within a given neural network is referred to as the network's *capacity*. Including too many parameters for a given modeling task may lead to a network fitting to the idiosyncrasies and noise within the training data and generalise poorly to unseen data,a phenomenon known as *overfitting*. On the other hand, a lack of training parameters leads to a very coarse approximation of the true function, which is known as *underfitting*. Including additional hidden layers increases the depth of a neural network, resulting in more abstract high-level representations being produced. This is where the phrases *deep learning* and *deep neural networks* come from.

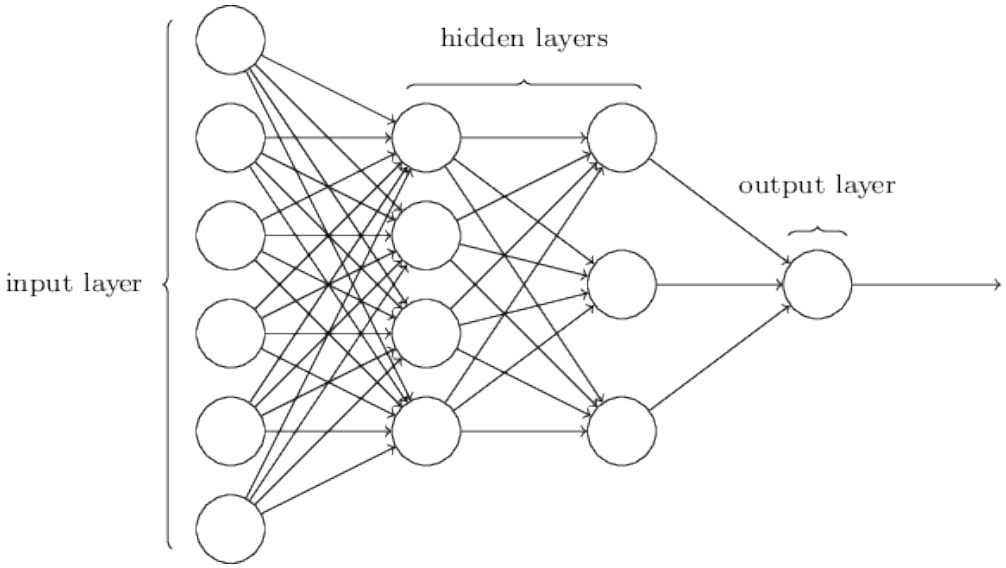

Figure 2.2: Multi-layer neural network.

2.1.2 Gradient Descent

Gradient descent is a first order optimisation algorithm used to final a local minimum for a given function in an incremental manner, even if the function is non-convex. This algorithm

can be used to optimise a given neural network by finding a set of model parameter values that minimises an objective function for a set of training samples. Gradient descent is based on the observation that if a multi-variate function $F(x)$ is defined and differentiable in the neighborhood of a point p, then $F(x)$ decreases *fastest* in the direction of the negative gradient of $F(x)$ at p. During each optimisation step the model parameters W are updated to move in this downward direction within the parameter space by subtracting the calculated gradient of $F(x)$ from the current parameter set, as described below:

$$W^{'} = W - \alpha \nabla_W F(x) \tag{2.4}$$

A scalar weighting α, refereed to as the learning rate, is applied to the subtracted gradient. The learning rate influences the size of the jump taken in a given optimisation step. After each iteration the gradient of $F(x)$ is re-calculated and another step is taken in the direction of the negative gradient. This concept is visualised in figure 2.3. Selecting an appropriate learning rate α and adjusting it during optimisation to ensure a local minima is converged upon is an active area of research within the machine learning community.

Gradient descent requires the objective function to be calculated using all training samples employed, which for large datasets may be highly computationally demanding. To address this issue representative mini-batches are randomly drawn from the training set in what is referred to as *Stochastic Gradient Descent (SGD)* (LeCun et al., 1988). The size of the mini-batch used has a strong influence on how quickly and smoothly the optimiser converges on a local minimum. Gradient descent can be efficiently carried out in deep neural networks by combining it with backpropagation (Rumelhart et al., 1986), an approach which re-uses gradients computed in higher layers to update lower layer weights, rather than naively calculating separate gradients for each layer during each optimisation step. The use of backpropagation significantly reduces

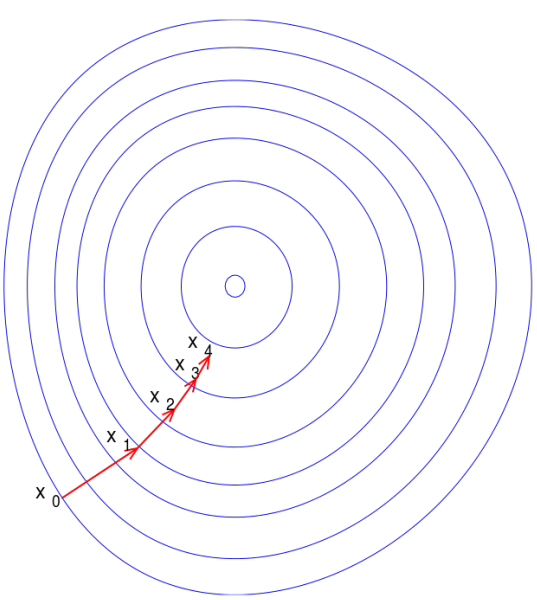

Figure 2.3: Example of gradient descent in action on a 2D plane. After each step the gradient is re-calculated and another step towards the local minima (bottom of the hill) is taken.

the computational complexity of gradient descent in neural networks, allowing optimisation to be performed on affordable hardware.

2.1.3 Convolutional Neural Networks

Neural networks are designed with vector input data in mind. This type of model does not scale well to multi-dimensional input data such as an images, which when flattened produce very large input vectors. These large input vectors require large numbers of weights per neuron in the input layer. For example, a 200×200 pixel RGB image requires 120,000 weights per neuron in the input layer when training a fully connected neural network. This high numbers of parameters will quickly lead to an overfit model.

This issue is addressed through the use of convolutions, resulting in so called *Convolutional Neural Networks* (CNN) being developed. A new type of network layer, referred to as convolutional layer is included. For the purposes of this explanation a 2-D network input is assumed

(i.e. an image). Each convolutional layer consists of a set of 2-D kernels (e.g. 3×3 pixels in size) which are convolved with an input to produce a set of feature maps. These feature maps highlight the occurrence of a given local feature within the input. Bias terms and activation functions are applied as in fully connected layers. This concept is visualised in figure 2.4. The input resolution can be reduced by performing a strided convolution, which skips every K pixels before convolving, producing a smaller feature map output. These layers can be stacked together and optimised via backprogation and gradient descent in the same way as fully connected layers. Max-pooling layers can also be interspersed within the network to reduce the resolution of feature maps further by taking the maximum value within each $J \times J$ region and discarding the rest.

After a series of convolutional layers a final set of lower resolution feature maps can be flattened to produce a 1-D vector and fed into a fully connected layer to perform a classification or regression task. Translational invariance is another advantage of convolutional layers. This type of approach can also be used to process other high dimensional modalities such as audio, sensor data and 3-D point clouds by altering the number of dimensions in the convolutional kernel.

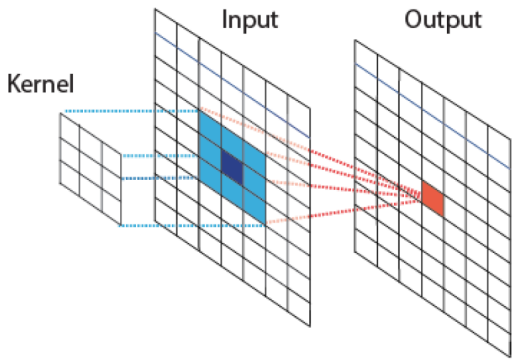

Figure 2.4: Example of a 2-D convolutional kernel in action

Other advancements in CNN architecture design have allowed for recursion to be included,

producing what is referred to as a recurrent neural network (RNN) (Funahashi and Nakamura, 1993). A recurrent neural network (RNN) is a class of neural network where connections between nodes form a directed graph along a sequence. This allows for the network to capture the temporal dynamics of sequential data. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. While RNNs approaches allow for long term temporal dynamics to be observed, they are designed to work for vectoral data and are thus not well suited for direct application to multi-dimensional video data.

Another development in the field is Generative adversarial networks (GANs). GANs are an approach to unsupervised machine learning, wherein a system of two neural networks contest with each other in a zero-sum game framework. They were introduced by Goodfellow et al. in 2014 (Goodfellow et al., 2014) and allow for improved dataset augmentation and pixel-wise processing tasks. This technique was not investigated directly as part of this study but is listed in the suggested future work.

2.1.4 Objective Functions

The objective function minimised when training a neural network typically consists of a loss term and one or more regularisation terms. A generalised form of this is given in equation 2.5, where W is the current set of network parameters, X is the set of training samples, L is the loss function and R is the regularisation term. The loss function measures the performance of the model on the training data, typically in terms of some error metric, while the regularisation term constrains the optimisation to add stability during training and prevent overfitting.

$$C(X;\Theta) = L(X;W) + R(W) \tag{2.5}$$

Loss Functions

For regression problems a mean squared error loss, given in equation 2.6, is typically used as the loss function L. N is the total number of training samples in a batch, $\hat{Y}(X_i; W)$ is the prediction for training sample i given the current set of training parameters W while $Y(X_i)$ is the corresponding ground truth value for sample i.

$$L(X, W) = \frac{1}{2N} \sum_{i=0}^{N} \left\| \hat{Y}(X_i; W) - Y(X_i) \right\|_2^2 \tag{2.6}$$

For multi-label classification problems a categorical cross entropy loss, given in equation 2.7, is typically used as the loss function L. N is total number of training samples, K is the total number of classes, $\hat{Y}_j(X_i; W)$ is the prediction score for concept j in training sample i while $Y_j(X_i)$ is the corresponding ground truth label.

$$L(X, W) = -\frac{1}{N} \sum_{i=0}^{N} \sum_{j=0}^{K} Y_j(X_i) \log(\hat{Y}_j(X_i; W)) \tag{2.7}$$

Regularisation

L_2 regularisation (Krogh and Hertz, 1992), often referred to as weight decay, is the most common regularisation term used for neural network optimisation and is described in equation 2.8. The thinking behind this regularisation term is to penalise large weight values. N is the total number of trainable weights in a given model, W_i is a given parameter and λ is the weight decay coefficient value. This coefficient value λ controls how severely large weight values are penalised during optimisation. Including this term prevents certain weights and therefore certain input values having a significant impact on the network prediction (i.e. overfitting to the training data). The choice of weight decay coefficient is another model selection issue presented when developing a neural network approach.

$$R(W) = \frac{\lambda}{2} \sum_{i}^{N} W_i^2 \tag{2.8}$$

2.1.5 Batch Normalisation

Batch normalisation (Ioffe and Szegedy, 2015) is a process used in neural networks to address an issue known as internal covariate shift. Covariate shift refers to the change in statistical distribution for the input x as the model is exposed to samples drawn from different domains. This change in distribution can make it difficult for a model to reliably converge on a local minimum, requiring normalisation to be applied to the input x. This issue is exacerbated in a deep neural network as the input distribution for each individual layer can vary and is affected by the parameters of the preceding layers. A small change to a given layer can affect the input distribution for several layers. This phenomenon is known as *internal* covariate shift. Therefore normalisation needs to be applied before each layer to produce a whitened distribution (i.e zero mean, unit variance) and ease model convergence.

Batch normalisation was developed to perform this during network training. It is referred to as *batch* normalisation because this process is performed in a batch-by-batch basis during stochastic gradient descent. At each step, a transform is applied to keep the mean close to zero and standard deviation close to 1 for a given layer input. The full algorithm is detailed in figure 2.5. A different normalisation transform is learned for each stage of the network. Applying batch normalisation when training allows for a higher learning rate to be employed due to the more stable convergence, enables deeper networks to be reliably trained, makes a given network more robust to variations in model initialisation and provides some additional regularisation which can improve overall predictive performance. A recent study from Santurkar *et al.* proposes that the true benefit of batch normalisation is the smoothing of the optimisation

landscape which increases stability and convergence rates. (Santurkar et al., 2018).

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x_i} \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x_i} + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Figure 2.5: Batch normalisation algorithm for processing the input x for a given layer.

2.1.6 Transfer Learning

Transfer learning, also known as inductive transfer, refers to the concept of taking knowledge gained while solving a given machine learning problem and applying that knowledge to a different but related problem. This obtained knowledge can include highly discerning feature detectors learned in convolutional layers and high level representations produced using fully connected layers that can be used to perform a range of analysis tasks.

Transfer learning in neural networks is performed by firstly initialising a new model using the parameters learned for a previous task. A subset of the weights within the new model can then be frozen during optimisation (no updates made) to fully retain the original knowledge and simplify the learning tasks by reducing the trainable parameter count. Choosing what weights

to freeze and what weights to fine tune from the original model is an issue actively researched by the machine learning community. Training a neural network using high level representations drawn from one or more frozen layers is referred to as *Feature Extraction*. A common approach to fine tuning a set of pre-trained layers involves training at a lower learning rate in order to retain most of the original knowledge and only subtly changing the learned function.

2.1.7 Optical Flow

Optical flow is the pattern of apparent motion of objects, surfaces and edges in a video caused by the relative motion between an observer and a scene (Gibson, 1950). Techniques have been developed to model this phenomenon and calculate the local motion vectors observed within a sequence of images. These local motion vectors can then be used to localise and subseqently recognise activity in video sequences. This data can also be used to perform other video processing tasks including video stabalisation and structure from motion approximation. An example of the these local motion vectors being estimated is presented in figure 2.6.

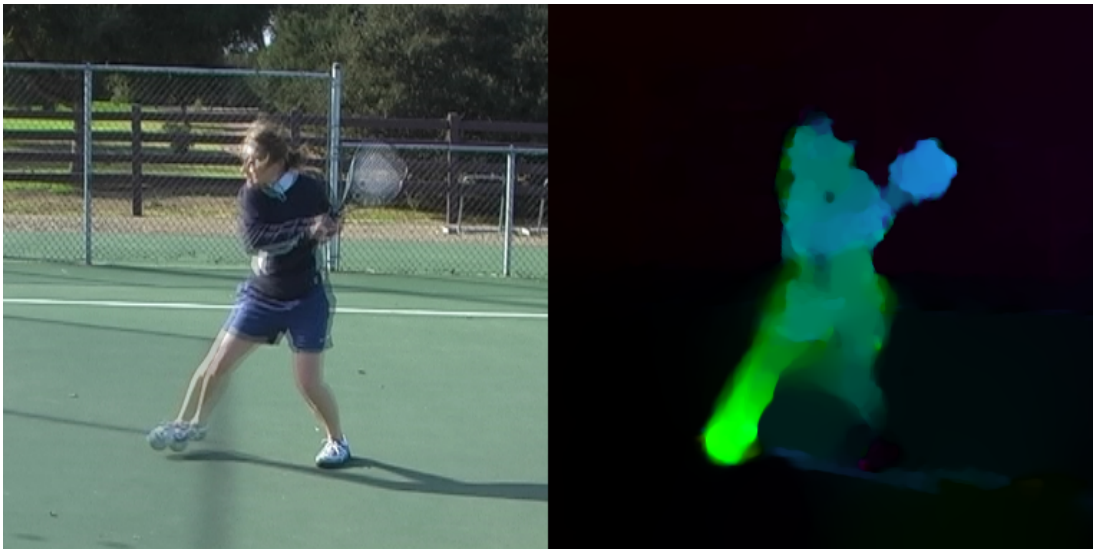

Figure 2.6: Local motion vectors captured through optical flow estimation. These local motion vectors are coloured to indicate the direction of motion.

When modelling optical flow there are several assumptions made: 1) the pixel intensities of an object do not change between consecutive frames 2) neighbouring pixels have similar motion patterns. Consider a pixel $I(x, y, t)$, taken from a given frame t in a video sequence. This pixel is displaced by (dx, dy, dt) in the next frame. If we assume the pixel intensity values of objects do not change between frames then the following can be defined:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{2.9}$$

Assuming the motion between frames is small, a Taylor series approximation can taken for the right hand side. Removing the common terms and dividing both sides by dt results in following:

$$f_x u + f_y v + f_t = 0 \tag{2.10}$$

This is referred to as the optical flow equation. f_x, f_y and f_t are the image gradients between the two frames and are defined as $f_x = \frac{\delta f}{\delta x}$, $f_y = \frac{\delta f}{\delta y}$ and $f_t = \frac{\delta f}{\delta t}$ respectively. These values can be calculated in a straightforward manner. On the other hand u and v represent the local motion vector of this pixel, defined by $u = \frac{dx}{\delta t}$ and $v = \frac{dy}{\delta t}$. The exact values of u and v are unknown and with a single equation cannot be easily calculated. Therefore several methods have been developed to approximate these values including the work of Gunnar Farneback (Farnebäck, 2003). It can be extremely difficult to accurately approximate optical flow when there is significant motion and illumination changes between frames. Estimating optical flow for the entirety of a high resolution video can be very computationally demanding and this has led to specialised hardware being developed for this task. Convolutional neural network approaches have also been developed to perform optical flow estimation, allowing hardware acceleration to be employed (Ilg et al., 2017).

2.1.8 Performance Metrics

Various performance metrics are used to benchmark machine learning algorithms for regression and classification tasks.

Regression

Mean absolute error (MAE) and root mean squared error (MSE), given in equations 2.11 and 2.12 respectively, are used to measure the performance of regression tasks. Y is the set of predicted values in the validation/test set while $\hat{Y}$ is the set of corresponding ground truth values.

$$MAE(Y, \hat{Y}) = \frac{1}{N} \sum_{i=0}^{N} \|Y_i - \hat{Y}_i\| \tag{2.11}$$

$$MSE(Y, \hat{Y}) = \sqrt{\frac{1}{N} \sum_{i=0}^{N} (Y_i - \hat{Y}_i)^2} \tag{2.12}$$

Classification

Accuracy is the most common metric used for multi-class classification problems and can simply be defined as the percentage of correct predictions made within a validation/test set. For neural networks the prediction made for a given input x is the class which has the highest likelihood score in the output y. Top-K accuracy is a variant of conventional accuracy where a prediction is deemed correct if the true class is within the top k likelihood scores predicted for a given input x. Top-K accuracy is often used in conjunction with accuracy to give a broader impression of the model's performance.

Binary classifiers can be evaluated using accuracy if a threshold is firstly applied, however this approach produces a very limited view of the model's performance, obscuring how the performance varies as this threshold is altered. Receiver operating characteristic (ROC) curves

were developed during the second world war to address this issue. An ROC curve plots the true positive rate (TPR) and false positive rate (FPR) of a binary classifier as the decision threshold is varied. A sample ROC curve is shown in figure 2.7. This method allows for a much broader comparison to be made between binary classifiers and removes the need for a threshold to be selected during evaluation. The overall performance of an ROC curve can be summarised into several scalar metrics, namely the area-under-the-curve (AUC) and equal-error-rate (EER). AUC can be calculated by taking the integral of the generated ROC curve and can be interpreted as the probability that the classifier in question will produce a higher likelihood score for a randomly chosen positive sample than a randomly chosen negative sample. The EER of an ROC curve corresponds to the value where the false negative rate and true positive rate are equal, with a lower value corresponding to a more robust classifier. EER is also a useful tool in selecting a threshold with the best tradeoff between true and false positives. Both EER and AUC are visualised in figure 2.8.

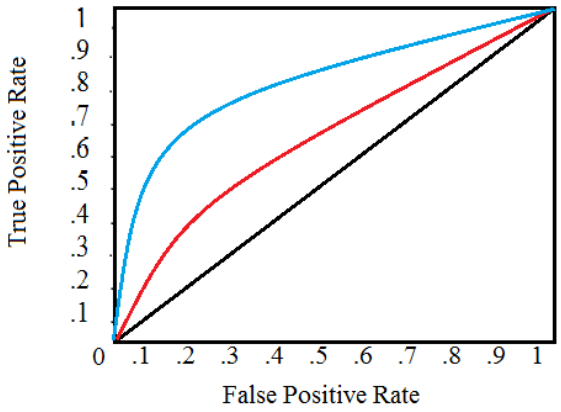

Figure 2.7: ROC curves for two binary classification systems (red and blue). The black line represents the performance achieved by a set of random guesses. Any curve below the black line is therefore deemed to be inferior to randomly guessing.

Precision and recall are performance metrics typically used in information retrieval problems. Precision refers to the fraction of correct samples among the retrieved samples while

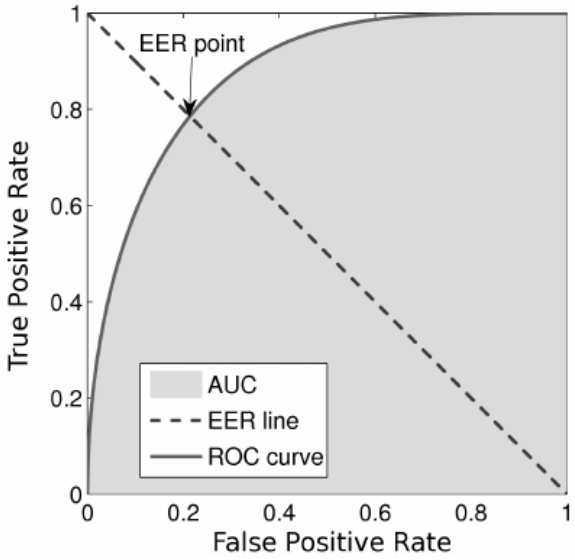

Figure 2.8: EER and AUC for a given ROC curve.

recall refers to the fraction of relevant samples retrieved from the total number of relevant samples in the collection. These metrics can also be computed for concept detection problems in fields such as computer vision, but require a detection threshold to be decided upon when using techniques such as neural networks. To avoid this threshold selection issue during evaluation the decision threshold is varied and the precision and recall values (P_n and R_n) are computed for all thresholds, producing a precision-recall curve showing how both metrics vary with the threshold. Average Precision (AP), given in equation 2.13, summarises a precision-recall curve in the same way AUC summarises an ROC curve. AP is calculated by producing a weighted mean of the precision values achieved for all thresholds, with the increase in recall from the previous threshold used as the weight in each case. Mean Average Precision (MAP) can then be calculated for a range of queries by taking a mean of the AP score calculated for each.

$$AP = \sum_{n=0}^{N} P_n(R_n - R_{n-1}) \tag{2.13}$$

2.2 Crowd Behaviour Analysis

Vision-based crowd behaviour analysis can be cleanly divided into two distinct tasks: crowd behaviour recognition and crowd behaviour anomaly detection. Behaviour recognition techniques attempt to categorise the collective behaviour observed in an image or video of a crowded scene, while behaviour anomaly detection methods are used to detect unusual behaviour which strays from an established norm.

2.2.1 Behaviour Recognition

Crowd behaviour recognition can be approached either as a single-label or multi-label problem. Single-label approaches categorise the collective behaviour into one of K mutually-exclusive classes, while multi-label approaches produce likelihood scores for a set of K behaviour concepts that can occur simultaneously. Behaviour concepts that are typically detected by these systems include violence, panic, running, standing and sitting. The typical pipeline for this task first extracts some local or global features from a given image or video, these features are then collated into a high-level descriptor before a classification step is performed. This generic pipeline can be used to describe most crowd behaviour recognition models and is illustrated in figure 2.9.

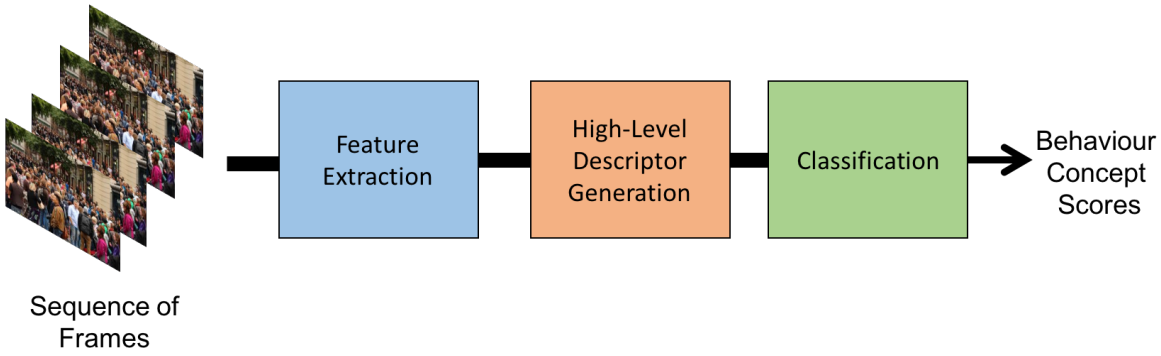

Figure 2.9: Crowd behaviour recognition pipeline

Local motion vectors generated via optical flow estimation have been used extensively for crowd behaviour recognition. An example of this is the video-level descriptor of Hassner *et al.* which extracts regional histograms of optical flow magnitude change and uses this descriptor to train a behaviour classifier (Hassner et al., 2012*a*). Local motion vectors were then combined with shape and appearance features for crowd behaviour recognition by Xu *et al.* who utilised the moSIFT descriptor and sparse feature encoding to achieve highly accurate violence recognition (Xu et al., 2014). Senst *et al.* propose the combination of Lagrangian direction fields with bag-of-words encoding to classify crowd behaviour, further improving violence recognition performance over the existing techniques (Senst et al., 2017) .

Multi-label behaviour recognition was firstly tackled by Shao *et al.* who trained a deep convolutional neural network to produce likelihood scores for 94 crowd behaviour concepts using both RGB and optical flow input channels (Shao et al., 2015). This work was then improved upon in a subsequent paper in which the authors anaylsed the crowd behaviour within a 3-D video volume by applying 2D convolutions across 3 orthogonal planes and fusing the results (Shao et al., 2016)

The leading approaches to crowd behaviour recognition tend to have a few key similarities: 1) they combine local motion and appearance features 2) they capture medium and long term trends observed over many frames 3) they produce a highly abstract and transform invariant representation of crowd behaviour before classification is carried out. Deep learning techniques, specifically convolutional neural networks have not been fully investigated for crowd behaviour recognition, particularly for violence recognition which has typically been approached using established hand-crafted features. This thesis attempts to define a set of best practices for vision-based crowd behaviour recognition.

2.2.2 Behaviour Anomaly Detection

Crowd behaviour anomaly detection methods attempt to classify whether a frame or sequence of frames contains unusual crowd behaviour. Unusual crowd behaviour is deemed to be that which strays significantly from an established norm within a given context. This type of approach is used to detect abnormal events that are either difficult to define or rarely occur, leading to a lack of training samples.

Early approaches such as the work of Boiman and Irani attempt to reconstruct a frame or spatio-temporal region using a codebook of *normal* behaviour samples for local frame regions, with a high reconstruction error corresponding to an anomalous region due to the difficulty in reconstruction (Boiman and Irani, 2007). This concept was refined by Roshtkhari and Levine who sorted spatio-temporal cuboid samples into a large contextual graphs before constructing a hierarchical codebook of *normal* behaviour samples (Roshtkhari and Levine, 2013). Mehran et al. proposed the social force model for behaviour anomaly detection (Mehran et al., 2009) Lu *et al.* proposed a more computationally efficient approach to reconstruction-based anomaly detection where a low rank projection captures the intrinsic compositions of small video regions (Lu et al., 2013). Reddy *et al.* developed an approach to anomaly detection which combines local motion and texture features with an efficient implementation of kernel density estimation (Reddy et al., 2011). Deep learning techniques were combined with a one-class SVM by Xu et al. to perform offline behaviour anomaly detection with a high degree of accuracy (Xu et al., 2015). Ravanbakhsh et al. propose the use of generative adversarial networks for behaviour anomaly detection (Ravanbakhsh et al., 2017)

Overall there has been minimal investigation into the use of deep learning techniques for crowd behaviour anomaly detection. Many of the existing approaches utilise highly engineered hand crafted features to localizing small unexpected objects within innocuous scenes rather

than detecting truly salient and noteworthy events in large scale video datasets. This thesis will investigate the use of deep learning for crowd behaviour anomaly detection.

2.3 Crowd Congestion Analysis

Crowd congestion analysis can be divided into two distinct tasks: crowd counting and crowd density level estimation. Crowd counting methods produce an estimate of the true number of people present in a scene. Crowd density level estimation (DLE) methods on the other hand approximate the congestion level observed within a scene and express it on a discrete scale (0-N). Crowd DLE can be viewed as a coarse but computationally efficient approximation of crowd counting.

2.3.1 Crowd Counting

Crowd counting can be approached either as an object detection problem, a count regression problem or as a heatmap generation task. The existing counting approaches are divided into these three categories.

Counting By Detection

Counting by detection approaches train an object detector to locate each individual person in a scene, typically using a key feature such as their head or torso. Examples of detection-based approaches include the head detector based method of Li *et al.* (demonstrated in figure 2.10) (Li et al., 2008), the multiple body part detector of Wu and Nevatia (Wu and Nevatia, 2005) and the marked point process (MPP) based approach to whole body detection from Ge and Collins (Ge and Collins, 2009). The accuracy of this type of approach suffers significantly from visual occlusions, resulting in rapid performance degradation as a crowd becomes highly congested

(i.e. several hundred people in frame).

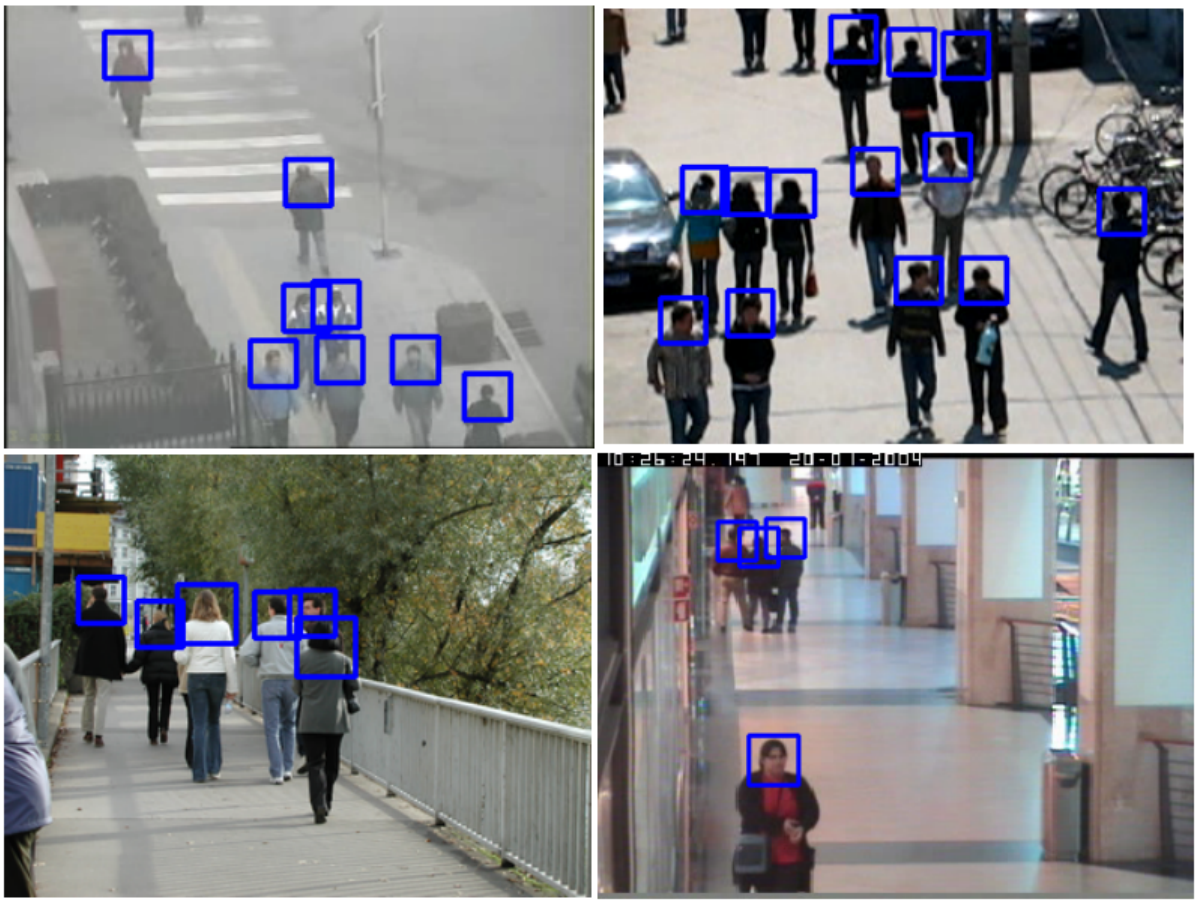

Figure 2.10: The head detector based crowd counting algorithm of Li *et al.* (Li et al., 2008)

Counting By Regression

Counting by regression methods on the other hand attempt to learn a direct mapping between a low-level representation (e.g raw pixel values) and the overall number of people within a frame or frame region. Individual people are not explicitly detected or tracked in these approaches. The holistic nature of this type of approach reduces the impact of visual occlusions on counting accuracy. Examples of regression-based counting include the work of Chen *et al.* who trained a count regressor by fusing local and global scene features (Chen et al., 2012) and the patch based neural network regressor of Han *et al.* (Han et al., 2017). While occlusion is less of an issue, regression-based techniques can often suffer from overfitting due to a lack of varied training

data and model regularisation during training.

Counting By Heatmap Generation

Finally, heatmap generation approaches tackle the counting problem by training a model to transform an image of a crowded scene into a density heatmap highlighting the locations of people within the scene. The produced heatmap is then integrated to produce an estimate of the true number of people. This concept is illustrated in figure 2.11. Ground truth heatmap images used during training are generated using manually produced head annotations. Gaussian blurring is typically applied to the produced heatmap to ease the learning task, while retaining the original count sum. Zhang *et al.* developed a dynamic blurring approach that applies more Gaussian blurring to heavily congested parts of the scene, leading to improved counting performance (Zhang et al., 2016). This dynamic blurring approach has been utilised by many subsequent approaches, however the generation algorithm features several hyperparameter choices that have not been validated experimentally or justified mathematically, leading to the possibility that they have been optimised specifically for a given test set. Other heatmap based approaches include the dilated convolutional neural network based approach of Li *et al.* who achieve high benchmarking performance but tune certain parameters for certain test sets, which limits the generalisation of their approach (Li et al., 2018).

Some novel approaches to heatmap based crowd counting switch between several individually trained CNN models following a density classification step (Sam et al., 2017; Wang et al., 2017) in an attempt to boost performance through density level specific estimators. While this approach leads to marginal performance improvements it results in a significant increase in training time and the number of model parameters required, limiting the scalability of the technique. Sindagi and Patel on the other hand attempted to jointly train a CNN regressor and

Figure 2.11: Crowd density heatmap example. The jet colourmap has been applied to the density heatmap (right). The integral of this ground truth image corresponds to the number of people present in this original image (left).

heatmap generation model whose outputs are fused to produce a final estimate (Sindagi and Patel, 2017).

Overall there has been a healthy level of research into machine learning based crowd counting with several approaches utilising deep learning techniques. Many of the developed approaches are slight variations on a core heatmap generation/regression concept with incremental performance boosts. Counting accuracy is shown to improve when training is carried out on smaller image regions in which pedestrians are observed to be more homogeneous in size. This eases the learning task significantly. It still remains unclear what the optimal implementation of deep learning techniques for crowd counting is, both in terms of model accuracy and efficiency. This thesis aims to identify these best practices for deep learning based crowd counting.

2.3.2 Crowd Density Level Estimation

Crowd density level estimation (DLE) methods attempt to classify the congestion level observed in an image of a crowded scene on a discrete scale (0-N). The main considerations with this task are the semantic meaning of the various density levels and the choice of what annotation scheme

to employ. Crowd DLE can be addressed either as a conventional classification problem, a regression problem with rounding applied or as an ordinal regression problem. Ordinal regression can be viewed as something of a hybrid between regression and classification models where distinct classification labels are produced but where the order and distance between labels is taken into account during optimisation.

An early approach to this task assigns 1 of 5 density levels to a scene based on the Minkowski fractal dimension (Marana et al., 1999). The density level labels used for this approach are inferred from the crowd count of each image, which removes subjectivity, but the density level labeling scheme used is far too granular and centered near 0 (the maximum density level of 5 refers to any image containing 60 or more people). This scheme cannot be applied in a meaningful way to highly congestion scenes with hundreds of people in frame. Xiaohua *et al.* used wavelet features to train an SVM to classify a crowded scene into 1 of 4 density levels (Xiaohua et al., 2006). Again the annotation scheme used is not comprehensive in terms of the congestion levels it covers and not consistent with other techniques. The commonly adopted SIFT local descriptor was used for density level estimation as part of a bag-of-words approach by Zhang and Zhang, classifying between *low*, *medium* and *high* density scenes (Zhang and Zhang, 2015). A CNN-based approach to density level estimation was developed by Fu *et al.* but this technique fails to even use a consistent density level scheme across the 3 test sets used for evaluation (Fu et al., 2015).

Overall the lack of an accepted benchmarking activity or a standardised density level annotation scheme are major obstacles to carrying out research for this task. New techniques cannot be quantitatively compared to older methods if the evaluation process is not consistent and clearly defined. One possible solution may be to re-purpose the leading crowd counting benchmarks that have already been accepted by the community and develop a standard crowd density level

annotation scheme based on the contents of these datasets. Using a broad set of images ranging in congestion from tens of people to thousands will allow for a more comprehensive image labeling scheme to be produced. This thesis will address the lack of a high quality dataset for this task and investigate the use of deep learning methods for density level estimation.

2.4 Multi-Task Learning

It has been empirically demonstrated that the predictive performance of supervised machine learning models can be improved by jointly training several related tasks at once (Caruana, 1998). Rich Caruana describes this concept of multi-task learning (MTL) as follows "Multi-task Learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias. It does this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better" (Caruana, 1998).

The benefits of multi-task learning have been successfully demonstrated in computer vision problem domains such as facial analysis (Ranjan et al., 2017), head pose estimation (Yan et al., 2016), medical imaging (Zhang et al., 2012) and non-vision tasks such as speech recognition (Seltzer and Droppo, 2013). Multi-objective approaches to crowd analysis have shown some initial promise, such as the work of Hu *et al.*, who included a density level estimation output to increased the robustness of their crowd counting model (Hu et al., 2016). To date, no crowd analysis model has been developed which jointly performs behaviour and congestion analysis tasks.

The multi-task learning concept has been applied to a variety of machine learning model types including neural networks. A neural network can be extended to perform multiple tasks by including additional output neurons to produce all of the required task predictions (Caruana,

1998). The internal representations generated by the hidden layers of a neural network are then shared between all tasks. An example of a multi-task neural network is shown in figure 2.12. MTL neural networks can be optimised via gradient descent by including a seperate loss term for each task in the overall objective function. Various combinations of loss functions can be used. Task specific weightings can also be applied to loss terms in the objective, giving certain task greater influence during optimisation. The MTL concept has also been applied to support vector machines, leading to improved performance on real and synthetic datasets (Evgeniou and Pontil, 2004). MTL extensions has also been developed for K-nearest neighbour (KNN) classification (Caruana, 1998) and decision trees (Kocev et al., 2007).

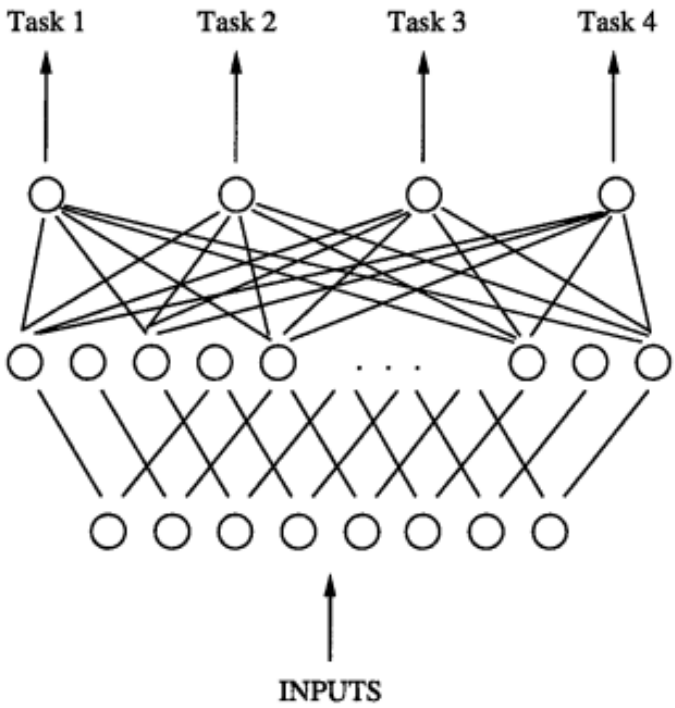

Figure 2.12: A multi-task neural network performing 4 tasks simultaneously for a given input x. The internal representations of the hidden layer are shared between all tasks.

Recently the MTL concept has been applied to very deep convolutional neural networks, allowing for highly abstract representations to be shared between tasks in areas such as facial analysis (Ranjan et al., 2017). One issue to consider when developing a multi-task CNN model

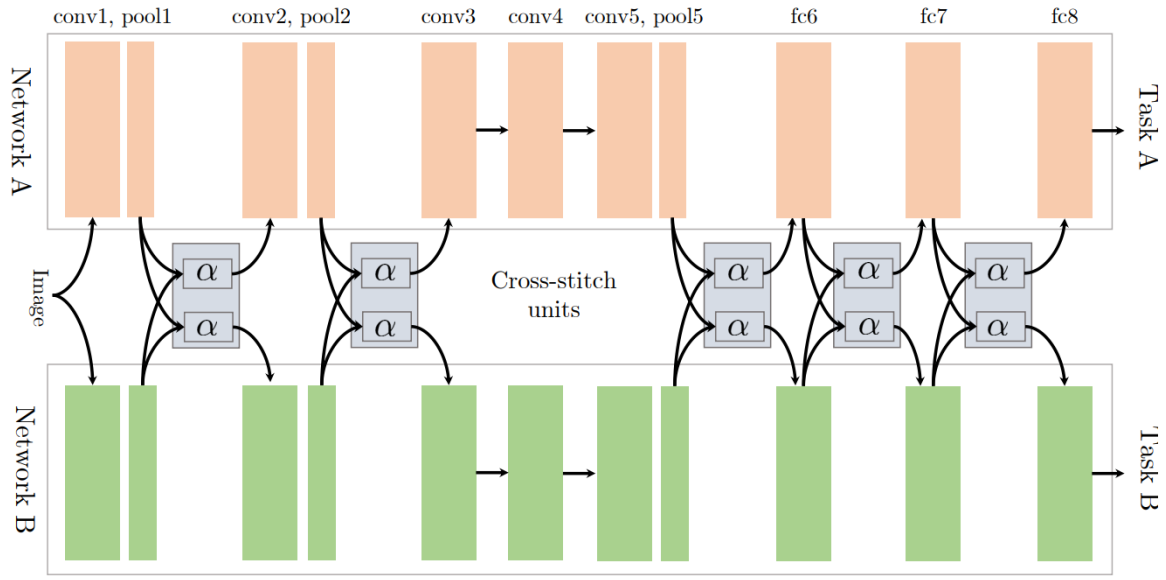

Figure 2.13: Cross-stitch unit of Misra *et al.* (Misra et al., 2016) applied to the Alexnet architecture (Krizhevsky et al., 2012). These additional units optimise the proportion of shared and task specific parameters in the network.

is the choice of what portions of the overall network should be shared between tasks and what should be task-specific. These choices can require extensive model validation and experimentation for each analysis problem tackled. Misra *et al.* propose the cross-stitch unit (illustrated in figure 2.13) to address this architecture selection issue (Misra et al., 2016). These cross-stitch units allow for a multi-task CNN to learn the optimal combination of shared and task specific representations throughout the network while training for multiple tasks in an end-to-end fashion. The main drawback of the cross-stitch unit approach is that it fails to reduce the overall parameter count across tasks. While the overall network of Misra *et al.* is fixed from the start, Lu *et al.* propose a multi-task architecture that widens dynamically during training and increases the level of representation sharing for similar groups of tasks (Lu et al., 2017). Kendall *et al.* take an alternate approach to multi-task CNN design by modeling the uncertainty of each task and altering the loss weighting of each task during training (Kendall et al., 2017).

Overall there has been significant investigation into MTL in deep neural networks with

improved performance observed in several computer vision problems with the inclusion of additional learning objectives. A range of approaches to automating multi-task CNN design have been developed, improving benchmarking performance and speeding up development time. There has, however, been very limited application of these techniques to vision-based crowd analysis, and no research into combining crowd behaviour and congestion analysis models. This previously under investigated area is addressed extensively in this thesis.

2.5 Domain Adaptation

Domain adaptation (DA) is a form of inductive transfer described by Gabriela Csurka as follows "Domain Adaptation (DA) is a particular case of transfer learning (TL) that leverages labeled data in one or more related *source* domains, to learn a classifier for unseen or unlabeled data in a *target* domain" (Csurka, 2017). A domain in this setting refers to a specific context with its own statistical distribution and specific traits. Examples of domains within computer vision include CCTV footage, medical imaging, facial analysis, and manufacturing. What distinguishes domain adaptation from multi-task learning is the assumption that the training samples from the source and target data are not available at the same time and thus the model must be training in a *sequential* manner, while MTL models are trained *simultaneously*. The distinction between MTL, DA and the various other forms of transfer learning are illustrated in figure 2.14.

The motivation for domain adaptation research is the observation that it may be infeasible to keep training data for a given task in a given domain indefinitely, due to storage costs, security and licensing reasons. However, the trained model produced using this training data is often significantly smaller and is likely to be kept and remain in use. Therefore the goal of DA is to utilise this model originally trained in a given source domain to produce a new model is a given target domain. The distinct statistical properties of the various domains, however, make this

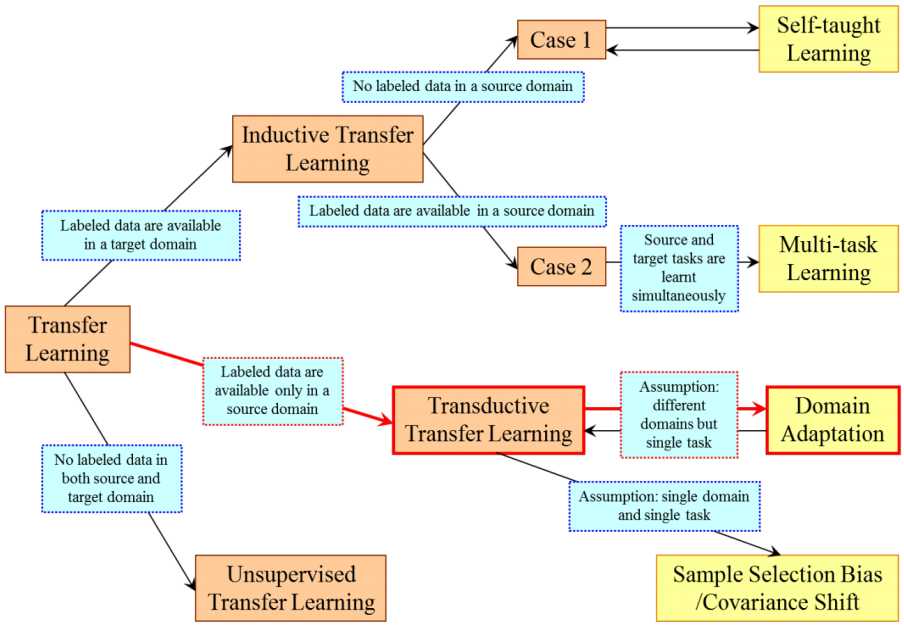

Figure 2.14: Overview of the various classes of transfer learning (Pan and Yang, 2010)

challenging to perform. Features learned in the source domain may not be highly discerning in the target. The ideal case for DA is to achieve strong predictive performance in the target domain while including a minimal amount of additional network parameters and maintaining the predictive performance originally achieved in the source domain, all in a singular model that can be extended over time.

Several domain adaptation strategies for vectorial data have been developed, often extending support vector machines to perform DA. The cross-domain SVM, proposed in (Jiang et al., 2008), constrains the impact of the source data by down-weighting support vectors from the source data that are far from the target samples. The Domain Transfer SVM of Duan *et al.* attempts to simultaneously reduces the mismatch in the distributions between two domains while learning a target decision function (Duan et al., 2009). DA has also been investigated in deep neural networks. A major obstacle to performing DA in deep CNNs is a phenomenon known as *catastrophic forgetting* (Goodfellow et al., 2013), where training a network to perform a new task results in the network *forgetting* the old task and changing the learned model weights

reached through optimisation, with an observable loss in predictive performance. This may not be an issue if all of the original models are retained indefinitely, but this results in a linear increase in the overall parameter count as new models are trained. This approach is not at all scalable and goes against the core philosophy of domain adaptation research. To avoid catastrophic forgetting, traditional transfer learning techniques such as feature extraction/weight freezing have been employed; however, this often leads to sub optimal predictive performance in the target domain.

To overcome these limitations several new approaches to CNN domain adaptation have been proposed. Li and Hoiem proposed the Learning without Forgetting (LwF) method for domain adaptation, which attempts to preserve source domain performance by initially storing the output of the source domain network for each sample in the target domain training set and optimising for these outputs as an auxiliary task (Li and Hoiem, 2017). This concept is illustrated in figure 2.15. Using this novel approach, LwF achieves superior source and target domain performance to feature extraction and fine-tuning approaches while introducing only a negligible amount of additional model parameters. LwF can be easily implemented on top of any existing network. Mallya *et al.* proposed the use of network pruning to free up model capacity during training and add new tasks in a sequential manner over time, all while achieving superior source and target performance to LwF (Mallya and Lazebnik, 2017). In a follow up paper the authors improve upon their results, using a technique they refer to as piggyback, which uses per task binary weight masks to *piggyback* new tasks onto an existing network (Mallya and Lazebnik, 2018). Rosenfeld and Tsotos perform DA by learning new CNN filters that are linear combinations of existing filters (Rosenfeld and Tsotsos, 2017). Rebuffi *et al.* developed the visual decathlon challenge for evaluating domain adaptation methods and proposed their own Residual Adapter Module (Rebuffi et al., 2017). This module performs DA by adding domain

specific normalisation and scaling throughout the network to adapt to the specific statistical distribution of each domain, all while maintaining original performance in the source domain and strong classification performance in the target with only a marginal increase in the overall parameter count. (Rebuffi et al., 2017).

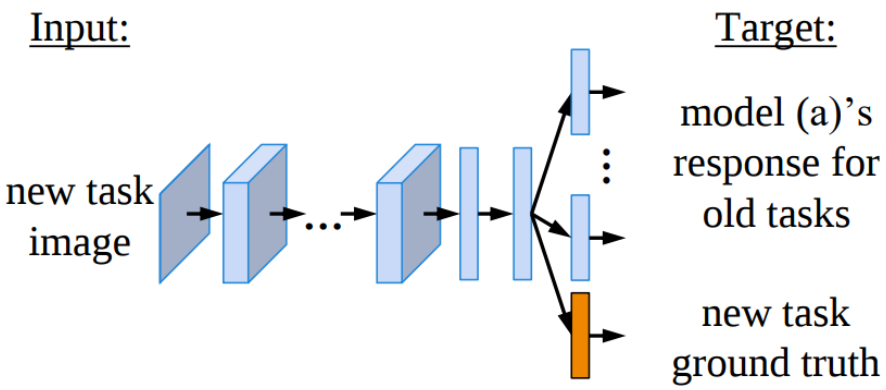

Figure 2.15: The learning without forgetting approach of (Li and Hoiem, 2017)

Overall there has been significant research into domain adaptation for computer vision tasks using neural networks. The most recent techniques have seen the field move beyond simple feature extraction and fine-tuning to more sophisticated methods with improved source and target domain performance. Generally these methods have been applied to commonly used image classification benchmarks while regression and object detection problems in areas such as crowd analysis and medical imaging have yet to be investigated fully.

2.6 Datasets

In this section the datasets used for the evaluation of various crowd analysis tasks are presented. The strengths and weaknesses of each dataset are highlighted before a final subset is decided upon for the experimental section of the thesis.

2.6.1 Crowd Behaviour Recognition

Violent-Flows

The Violent-Flows dataset of Hassner *et al.* contains 246 video clips of violent and non-violent crowd behaviour captured from CCTV cameras and mobile phones (Hassner et al., 2012*a*). This binary classification task is evaluated using a 5-fold cross validation with methods compared using mean accuracy and ROC curve AUC score. Classification is carried out at the video level. The clips in this dataset range in length from 1 to 6 seconds with a mean length of 3.6 seconds. The content of this dataset are illustrated in figure 2.16. This collection contains a variety of challenging real-world scenes and has received significant interest from the research community. Given the relatively small size of this dataset it is an ideal collection on which to develop a crowd behaviour analysis system before testing on larger datasets.

Figure 2.16: The Violent-Flows dataset (Hassner et al., 2012*a*). Bottom left: violent clips. Top right : non-violent clips

WWW Crowd Dataset

The WWW Crowd Dataset contains 10,000 clips of crowded scenes labelled for 94 crowd behaviour concepts and related scene content labels (Shao et al., 2015). WWW refers to *Who What Why* for this dataset. These labels are not mutually exclusive and thus this dataset is treated as a multi-label classification problem. These 10,000 clips are split into training, validation and test sets using a 7:1:2 ratio while methods are evaluated using mean AUC score and mean average precision (mAP). Classification is also carried out at the video level. This collection is illustrated in figure 2.17. The size and scope of this dataset make it ideal for evaluating a crowd behaviour recognition system in a comprehensive fashion. There has been limited usage of this dataset despite the high citation rate of the original paper, which may be due to the time-consuming nature of working with this 8 million frame collection.

PETS2009

The PETS2009 dataset (Dee and Caplier, 2010) has been utilised for a range of analaysis tasks including crowd behaviour recognition. 4 distinct crowd behaviour classes (running, loitering, dispersal, formation) are included in this collection. This dataset is quite limited in terms of variety and the crowd behaviour events are produced artificially. Classification is carried out at the frame level. Sample frames taken from these scenes are shown in figure 2.18

2.6.2 Crowd behaviour Anomaly Recognition

UMN Dataset

The University of Minenesota unusual crowd behaviour dataset [1] contains 11 video sequences filmed in 3 locations. Each sequence begins with a period of normal crowd behaviour before

[1] http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi

Figure 2.17: The WWW Crowd Dataset (Shao et al., 2015). Samples frames are shown as well as a word cloud presenting the distribution of the various scene concepts.

a panic event/anomaly occurs near the end. Anomaly detection is performed at the frame level and evaluated using AUC score. The events that take place in this dataset are artificial and the dataset only contains frames from 3 scenes. A sample frame is shown is shown in figure 2.19.

Figure 2.18: The limited range of scenes contained within the PETS2009 Dataset.

Figure 2.19: A sample frame taken from the UMN dataset.

UCSD Dataset

The UCSD Anomaly Detection Dataset (Mahadevan et al., 2010) contains 98 short clips taken from two scenes. Each scene is evaluated separately and divided into a set of normal behaviour training and test clips. Anomalies must be localised spatially as well as temporally for this benchmark. Events that are defined as anomalies for this dataset include people walking on the grass and unexpected objects such as cars entering the scene. The value of developing a technique to detect such innocuous events is questionable. This dataset is also extremely homogeneous in terms of scene content with only two locations used and a fixed camera angle for both. The one advantage of this dataset over UMN is that the *anomalous* events are naturally

occurring. A sample frame is shown in figure 2.20.

Figure 2.20: A sample frame taken from the UCSD Anomaly Detection Dataset (Mahadevan et al., 2010).

LV Dataset

The LV (Live Videos) behaviour anomaly dataset (Leyva et al., 2017) contains 4 hours of footage ranging dramatically in scene content, image resolution and the types of behaviour anomalies present. The dataset is broken up into 28 distinct sequences each beginning with a period of normal behaviour for model training followed by a test region in which one or more anomalies may occur. All events are naturally occurring and must be localised spatially and temporally within each test sequence. Techniques are evaluated in terms of AUC score. Sample frames from this collection are presented in figure 2.21. This collection is by far the most comprehensive and challenging crowd behaviour anomaly dataset available to the research community.

2.6.3 Crowd Counting

UCF_CC_50 Dataset

The UCF_CC_50 dataset (Idrees et al., 2013) contains 50 images of highly congested scenes fully annotated for crowd counting via dot maps. This highly challenging collection contains

Figure 2.21: Sample frames taken from the LV dataset (Leyva et al., 2017).

significant variation in scene content and is benchmarked on using a 5-fold cross validation. Performance is evaluated using Mean Absolute Error (MAE) and Mean Squared Error (MSE). It has been suggested in the literature that the annotations used for this dataset are not entirely accurate due to the extremely high number of people contained in each scene (Hu et al., 2016). Several images in this dataset contain over 4000 people and are captured at a resolution lower than 1080×1920 (HD). Without a fully reliable ground truth the value in benchmarking on this collection is limited. A sample image taken from this dataset showing the highly congested

crowds is presented in figure 2.22.

Figure 2.22: Sample image taken from the UCF_CC_50 dataset (Idrees et al., 2013).

ShanghaiTech Dataset

The ShanghaiTech dataset (Zhang et al., 2016) contains 1198 images of crowded scenes ranging in the number of people present from a few dozen to several thousand. This collection is split into a medium-high congestion set (referred to as part A) and a low-medium congestion set (referred to as part B). Part A images contain between 0-3000 people while Part B images contain between 0-600 people. These two sets are evaluated separately with their own train/test splits. Performance is again evaluated using Mean Absolute Error (MAE) and Mean Squared Error (MSE). It is clear when looking at this collection that the ground truth annotations are far more reliable and the level of variation in scene content is much higher than UCF_CC_50. Sample frames from parts A and B of the ShanghaiTech dataset are shown in figure 2.23.

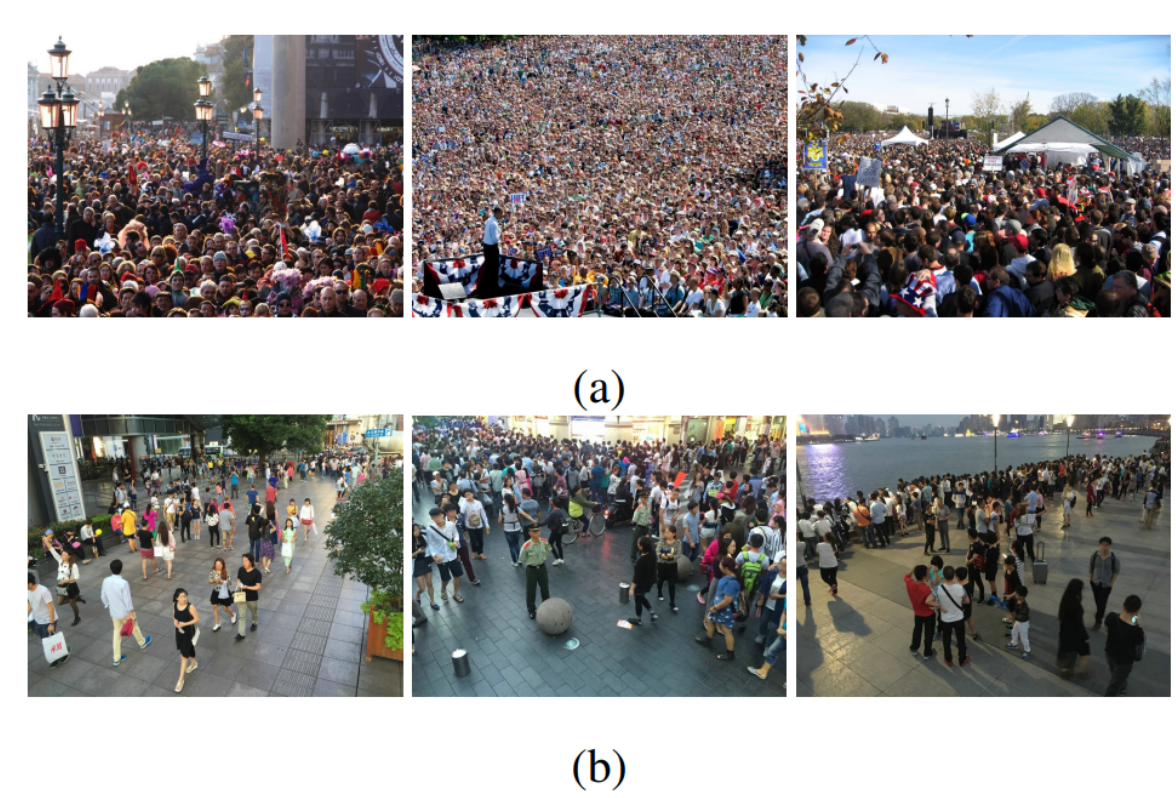

Figure 2.23: Sample image taken from the ShanhaiTech dataset (parts A and B) (Zhang et al., 2016).

UCSD Dataset

The UCSD crowd counting dataset repurposes the same set of clips used for the UCSD anomaly detection dataset by applying a set of head annotations (Mahadevan et al., 2010). This collections consists of 2000 images of low density scenes captured from just two cameras. The number of people present ranges from 11 to 46. The frames captured for this dataset are of a significantly lower resolution (158 $\times$ 238) than the ShanghaiTech and UCF_CC_50 sets.

2.6.4 Crowd Density Level Estimation

For the crowd DLE task there are no commonly accepted benchmarks and annotation schemes in use. There have been a few datasets utilised for this task, which vary in quality but in general lack the range in crowd congestion and scene content required to properly evaluate a given method.

PETS2009 Dataset

The PETS2009 dataset has been adapted for crowd density level estimation with a 4 density level scheme applied. Frames are taken from 5 distinct clips within the overall PETS2009 set. The 4-level annotation scheme covers only a small range of crowd count values, with the maximum level referring to any image with more than 21 people present. This type of annotation scheme is not appropriate for developing a robust and deployable density level estimator for all scenes and contexts.

Subway Dataset

The Subway sequence of Ma *et al.* has been used for density level estimation, with a 5-level scheme employed (Ma et al., 2008). Similar to PETS2009 this set consists entirely of low density images captured from a single camera location. For example the maximum density level for this benchmark corresponds to scenes with more than 31 people present.

2.6.5 Discussion

The quality of datasets used across the four crowd analysis tasks discussed ranges dramatically from thousands of fully annotated clips taken from hundreds of camera locations to a few hundred frames taken from a single location. For the experimental phase of this thesis the goal is to use challenging datasets that provide a broad and representative set of samples for fair evaluation of a method. Having a trusted set of annotations is also very important. With this in mind the following datasets will be used for experimentation:

- **Crowd behaviour Recognition**: Violent-Flows, WWW Crowd
- **Crowd behaviour Anomaly Detection** : LV Dataset

- **Crowd Counting** : ShanghaiTech
- **Crowd Density Level Estimation**: ShanghaiTech (adapted for DLE)

As there is no dataset of sufficient quality for the crowd DLE task, the ShanghaiTech collection is adapted for this task. The crowd count annotations used for this set can be easily converted to density level labels using an appropriate scheme (discussed fully in chapter 4). The significant range in congestion level observed in this collection makes it ideal for the evaluation of a DLE method.

2.7 Summary

This chapter began by presenting some important background theory relevant to the research work carried out in this thesis. Following this a review of the leading techniques for each of the four crowd analysis tasks was presented. Promising initial work in deep neural network based crowd analysis has been demonstrated for all tasks. Other prominent developments include the use of reconstruction error when detecting crowd behaviour anomalies and the observed superiority of patch-based training for crowd counting systems via heatmap generation. The evolution of multi-task learning strategies, particularly in the context of neural networks, was then discussed. While there have been many recent advancements within the area of MTL, these techniques have yet to be fully investigated for crowd analysis. The leading domain adaptation methods were then presented, with some methods adding auxiliary learning objectives to preserve performance in the source domain while others include domain specific normalisation and scaling. These techniques have been evaluated mainly on commonly used image classification benchmarks but have not yet been tested on regression and object detection tasks in areas such as crowd analysis and medical imaging. Finally, the leading publicly available datasets used for

crowd analysis were evaluated, with a subset decided upon for the experimental phase of this thesis.

Chapter 3

Crowd Behaviour Analysis Via Deep Neural Networks

3.1 Introduction

This chapter investigates the use of deep neural network techniques for the related computer vision tasks of crowd behaviour recognition and crowd behaviour anomaly detection. Research question 1 is addressed in this chapter as well as in chapter 4. Various convolutional neural network configurations, preprocesing steps and implementation strategies are evaluated for each task in an attempt to find the best practices for deep learning based crowd behaviour analysis. A baseline run using hand crafted features is also included for each task to compare performance with deep learning methods. Following the development of a refined method for each task using a validation set, comparisons are made with the leading techniques from the literature using a larger test set.

3.2 Contributions

The main contributions of this chapter are listed below:

- A 3D Late Fusion CNN approach is developed for crowd behaviour recognition;
- The trained model is used to perform distance-based crowd behaviour anomaly detection on the LV Dataset;
- The proposed technique is shown to be superior to a hand-crafted baseline for both behaviour recognition and anomaly detection;
- State-of-the-art performance is achieved on the LV and Violent-Flows datasets.

3.3 Experimental Framework

The proposed crowd behaviour analysis models are developed using a common framework in which certain hyperparamaters and model selection choices are kept consistent across all experiments. This limits the parameter space to explore during validation and focuses experimentation on the more impactful model selection issues.

3.3.1 Fixed Hyperparameters

Learning Rate

Learning rate selection issues often delay model development when optimising a neural network using Stochastic Gradient Descent (LeCun et al., 1988), requiring extensive validation and refinement for a given model and dataset. In order to alleviate these issues the Adagrad optimiser (Duchi et al., 2011) is used for all experiments in this chapter. Adagrad adapts the learning rate

dynamically for each weight individually in an attempt to remove global learning rate selection issues and has been shown to reliably converge to local minima (Dean et al., 2012).

Network Architecture

There have been numerous convolutional neural network architectures proposed in the literature (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015; He et al., 2016; Howard et al., 2017) with improvements in predictive performance, accelerated convergence rates and reductions in the overall parameter count observed for various designs. The focus of this thesis, however, is not implicitly in CNN design and therefore the model selection space must be reduced in order to focus on the more relevant design choices for crowd analysis.

To this end, the Resnet family of CNN architectures are chosen for model development (He et al., 2016), specifically the 18 layer and 50 layer variants. A Resnet architecture consists of a series of residual blocks, illustrated in figure 3.1, each containing several convolutional layers, a feedforward connection and an element-wise addition. This feedforward connection results in a residual representation being produced by each block. These residual representations are empirically shown to allow for faster convergence of very deep networks. Rectified linear unit activations and batch normalisation (Ioffe and Szegedy, 2015) are applied after each conovolutional layer. Following a series of residual blocks an average pooling step is carried out before the set of feature maps is flattened and a single fully connected layer is used to produce a network output. This type of architecture can easily be made deeper by including more residual blocks or more convolutional layers per residual block. Strong benchmarking performance has been demonstrated for various tasks using Resnet (He et al., 2016). The exact configurations of the Resnet18 and Resnet50 networks are presented in figure 3.2 including the kernel size, number of kernels and number of residual blocks in each stage of the network.. Max-pooling is

performed after conv3_1, conv4_1, and conv5_1 with a stride of 2.

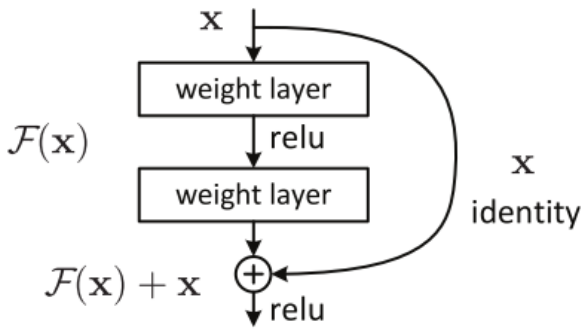

Figure 3.1: Residual block used in the architecture of (He et al., 2016).

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 3.2: Details for the Resnet family of architectures (He et al., 2016).

Model Regularisation

L_2 regularisation is used for all experiments. A λ value of 0.001 is employed to train the Resnet architecture in use just as it is in the original Resnet paper (He et al., 2016).

Data Augmentation

The following data augmentation steps are performed to increase the variation of training images and boost model generalisation:

- Random spatial cropping (aspect ratio preserved)
- Random horizontal flips
- Overlapping temporal crops in video (starting from every 5th frame)
- Dataset shuffled after each training epoch

All random cropping and flips are performed dynamically during training. Fixed length temporal crops are taken when using a multi-frame technique, with the starting point of each temporal crop 5 frames apart. The length of the temporal crop taken is experimented with using validation sets. At inference time, centre crops are taken for each frame, again with the aspect ratio preserved and any necessary downsampling performed to best fit the network input size. Vertical rotations are not applied as this is deemed to change the semantic meaning of a crowd video.

Model Initialisation

Network parameter initialisation is handled for all crowd behaviour analysis experiments via the uniform initaliser of Glorot and Bengio (Glorot and Bengio, 2010) while the bias terms are initalised to zero. The only time these initialisation steps are not carried out is when a pre-trained model is being used to initialise a given set of network layers. The pre-trained network used for all cases has been trained on the ImageNet ILSVRC collection (Russakovsky et al., 2015).

Loss Functions

For multi-label behaviour recognition problems, a binary cross entropy loss, given in equation 3.1, is minimised following a sigmoid activation on the final network output. K is the total

number of concepts, $\hat{S}_j$ refers to the predicted probability score for concept j while S_j refers to the ground truth score.

$$L_{\text{BCE}} = -\sum_{j=1}^{K} S_j \log(\hat{S}_j) + (1 - S_j) \log(1 - \hat{S}_j) \tag{3.1}$$

For single-label behaviour recognition problems a categorical cross entropy loss, given in equation 3.2, is minimised following a softmax activation on the final network output. K is the total number of concepts, $\hat{S}_j$ refers to the predicted probability score for concept j while S_j refers to the ground truth score. Both loss functions are calculated and summed for a batch of training samples before the model parameters are updated. In all cases the chosen loss term is combined with a regularisation term to form the overall objective function.

$$L_{\text{CCE}} = -\sum_{j=1}^{K} S_j \log(\hat{S}_j) \tag{3.2}$$

Hardware

The following hardware setup is used for all crowd behaviour analysis experiments: An Nvidia GTX 970 GPU with 4GB of VRAM is used for CNN optimisation and inference. An 8 core Intel i7-4790K CPU with 32GB of GDDR4 RAM is used for all dataset generation and experiment management tasks. While it has been suggested recently that smaller batch sizes generalise better (Masters and Luschi, 2018), large batches are required to perform batch normalisation effectively and therefore the maximum batch size possible using the available memory is used for each training experiment.

The common set of model selection parameters used for all experiments in this chapter are summarised in table 4.1.

Optimiser	Adagrad (Duchi et al., 2011)
CNN Architecture	Resnet (18, 50 layers)
Regularisation	L_2 Weight Decay (0.001)
Augmentation	Random Crops, Random Flips, Temporal Overlap (Video)
Initalisation	(Glorot and Bengio, 2010), bias terms set to 0
Loss Functions	BCE for multi-label, CCE for single-label
Hardware	4GB Nvidia GTX 970 GPU, 8 core Intel i7 CPU, 32GB RAM

Table 3.1: Common training framework used for all crowd behaviour analysis experiments

3.3.2 Model Selection Issues Investigated

With this experimental framework in place the focus of this chapter can now be discussed. The following model selection issues will firstly be investigated for crowd behaviour recognition:

- Model capacity (number of network layers)
- Training from scratch v. fine-tuning v. feature extraction
- Single frame v. multi-frame methods
- Optical flow input v. RGB input v. Fusion of RGB+OF

The following model selection issues will be investigated for crowd behaviour anomaly detection:

- Single-frame v. multi-frame features
- Optical flow input v. RGB input v. Fusion of RGB+OF
- Distance metric used for outlier detection

3.4 Crowd Behaviour Recognition

Crowd behaviour recognition methods are developed and benchmarked in this section using the Violent-flows and WWW Crowd datasets. These collections represent a small scale, single-label recognition problem and a large scale, multi-label recognition problem respectively. Both collections are labelled at the clip level. Performance is evaluated on the violent-flows dataset using mean accuracy and AUC score while AUC and mean Average Precision are used for the WWW Crowd collection. Accuracy cannot be used for the WWW crowd set due to it being a multi-label problem. One of the five validation folds that make up the violent-flows dataset is used for model selection experiments before a full five-fold cross validation is carried out to compare with the leading approaches. For validation experiments on a single fold no error margins can be computed. WWW Crowd on the other hand has its own dedicated validation and test sets. Optimisation is carried out for 25,000 iterations for all experiments. The corresponding number of training set epochs varies with the training set and the CNN model in use (which in turn affects the batch size). The size of the training, validation and test sets used for each dataset are listed in table 4.2.

Dataset	**Training Set Size**	**Validation Set Size**	**Test Set Size**
WWW Crowd	7249 Clips	917 Clips	1834 Clips
Violent Flows	200 Clip	50 Clips	50 Clips

Table 3.2: Training, validation and test set sizes for the WWW Crowd and violent-flows datasets. A 5-fold cross validation is carried out at test time for the violent-flows data set.

3.4.1 Hand-Crafted Baseline

In order to compare deep learning models with hand-crafted methods an implementation of Hassner *et al.'s* ViF descriptor (Hassner et al., 2012*a*) is produced and used as a hand crafted baseline run for crowd behaviour recognition. This method analyses the statistics of how optical flow vector magnitudes change over time, producing a fixed length descriptor for a variable length sequence. Classification is performed using a linear support vector machine. For multi-label problems a separate linear SVM is trained for each concept in a one-v-rest fashion.

3.4.2 Deep CNN Approach

Model Capacity, Trainable Parameters

This initial set of experiments use networks of various depths (Resnet18 and Resnet50) while also comparing various training strategies for each (training from scratch, fine-tuning and feature extraction). All 6 training permutations are evaluated on both the violent-flows and WWW crowd validation sets. For all runs a single frame classification model is trained, with the mean frame level predictions across the entire clip used for evaluation. The final fully connected layer of a given network is adjusted to produce the appropriate number of classes for each dataset (94 for WWW Crowd, 2 for Violent-Flows). Raw RGB frames are used as the input for all runs. The results of these experiments are presented in table 3.3, with each unique approach given an identification number which are used throughout the chapter to avoid any ambiguity.

The best performing run on the violent-flows validation set employs feature extraction from a pre-trained Resnet18 model. For this small scale classification task a lower capacity model utilising an existing feature set resulted in superior accuracy and AUC scores. On the other hand the best validation performance on WWW Crowd was achieved when fine-tuning a Resnet50 model end-to-end. The larger and more varied dataset supplied with WWW Crowd takes ad-

vantage of the additional capacity of a full Resnet50 model. In all cases transfer learning from a pre-trained model resulted in superior performance and deep learning runs outperformed the hand crafted baseline. These optimal configurations for each dataset are the starting point for all subsequent experiments.

Approach	**VF: ACC**	**VF: AUC**	**WWW: mAUC**	**WWW: MAP**
1: Resnet18 From Scratch	85%	0.87	0.85	0.415
2: Resnet18 Fine-tuning	87%	0.9	0.892	0.462
3: Resnet18 Feature Extraction	**91%**	**0.94**	0.885	0.456
4: Resnet50 From Scratch	83%	0.85	0.865	0.434
5: Resnet50 Fine-tuning	85%	0.87	**0.902**	**0.468**
6: Resnet50 Feature Extraction	90%	0.85	0.896	0.462
7: ViF	82%	0.84	0.66	0.100

Table 3.3: Comparison of the various network architectures and training strategies for single-frame crowd behaviour recognition on the WWW Crowd and violent-flow validation sets. The hand-crafted baseline is also included for both datasets. Each approach to behaviour recognition is given a unique identification number which are used in all subsequent tables.

Single-Frame v. Multi-Frame Models

The next set of experiments compares the optimal single frame runs for each dataset with various multi-frame CNN video recognition techniques from the literature which are re-implemented as part of this work. These include the late fusion 3-D CNN approach of Carreira and Zisserman (Carreira and Zisserman, 2017) (detailed in figure 3.3), the fully 3-D CNN of Tran *et al.* (Tran et al., 2015) (detailed in figure 3.4) and the RNN/Long-Short Term Memory (LSTM) approach of Ng *et al.* (Ng et al., 2015) (detailed in figure 3.5). These methods allow for the temporal dy-

namics of crowd behaviour to be captured alongside the local appearance patterns. The results of this experiment are presented in table 3.4

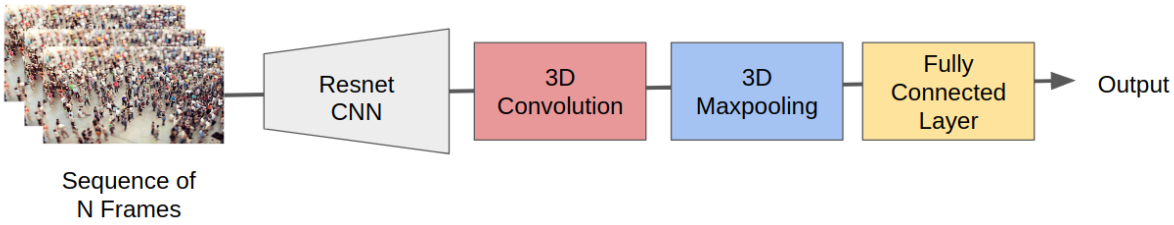

Figure 3.3: late fusion 3D CNN approach of Carreira and Zisserman (Carreira and Zisserman, 2017). 2D feature maps generated from each of the N frames ingested are fused along the temporal axis before a 3D convolutional layer, 3D maxpooling and fully connected layer are applied to produce a classification output. The 3D convolutional layer contains 256 kernels each $3 \times 3 \times 3$ in size.

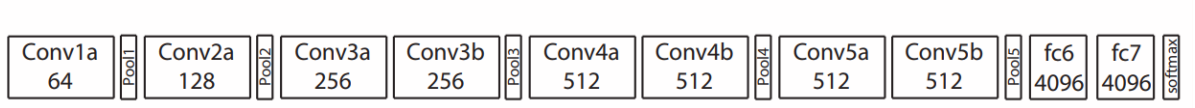

Figure 3.4: Fully 3D convolutional architecture of Tran *et al.* (Tran et al., 2015). All convolutional layers perform 3D convolutions with $3 \times 3 \times 3$ kernels and stride 1. The number of kernels contained in each layer is listed alongside the layer name. 5 frames are fused together along the temporal axis before being processed by this network. This network is trained from scratch due to the very distinct architecture.

The inclusion of temporal information results in noticeable improvements for the WWW Crowd dataset, with the *Late Fusion 3D Resnet18-FT* run producing the best validation performance. This late fusion technique did not result in any improvement for the smaller violent-flows dataset which does not contain enough training samples to train a high performing multi-frame behaviour recognition model. In its current configuration, the late fusion approach combines 5 frames spaced 10 frames apart (covering an overall 50 frame region). To investigate other configurations of this approach a further experiment is carried out which includes a wider temporal region. The spacing between extracted frames is increased to 15 and 20 for two additional runs. The results of this experiment are shown in table 3.5.

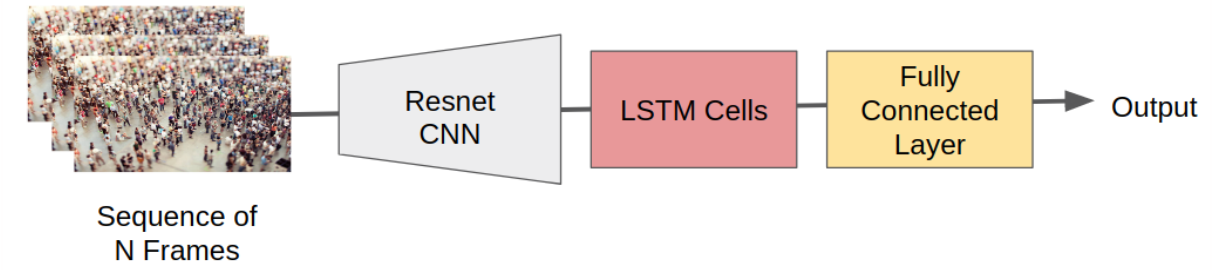

Figure 3.5: LSTM based video recognition architecture of Ng (Ng et al., 2015). Feature vectors are extracted from a sequence of N frames, fused temporally and passed through a deep LSTM block containing 5 layers of LSTMs, each with 512 memory cells. The output from the LSTM at the final time step is fed into a fully connected layer to produce a classification output.

Significant performance improvements are observed on the WWW crowd dataset as more temporal information is exposed to the network with the *5 frames, Spaced 20 Frames Apart* run producing the best overall validation scores on the WWW Crowd dataset. This is not the case for the violent-flows dataset, most likely due to the smaller training set which does not have enough variety to accurately model longer term behaviour patterns. Ideally more temporal information could be exposed to the network (e.g. all 100 frames within a 100 frame region) but the hardware setup used for these experiments cannot support the training of such a high capacity model. In the original paper for the late fusion 3D CNN model the authors use 64 GPUs in parallel (Carreira and Zisserman, 2017).

Optical-Flow Channels v. Raw RGB Channels

The final set of validation experiments for this task looks at the impact of using pre-computed optical flow channels instead of raw RGB channels for crowd behaviour recognition. All runs involve an identical late fusion 3D CNN architecture Carreira and Zisserman (2017), with 5 frames spaced 20 apart fusion applied in all cases and a Resnet18 CNN He et al. (2016) used

Model	VF: ACC	VF: AUC	WWW: mAUC	WWW: MAP
3: Resnet18-FE-SF	**91%**	**0.94**	0.885	0.456
6: Resnet50-FE-SF	85%	0.87	0.902	0.468
8: Late Fusion 3D Resnet18-FT	85%	0.87	**0.908**	**0.503**
9: Late Fusion 3D Resnet18-FE	89%	0.88	0.905	0.492
10: LSTM Resnet18-FT	83%	0.85	0.845	0.467
11: LSTM Resnet18-FE	85%	0.87	0.865	0.478
12: 3D CNN (From Scratch)	82%	0.85	0.815	0.423
7: ViF	82%	0.84	0.66	0.100

Table 3.4: Comparison of the various multi-frame approaches to crowd behaviour recognition with a set of single-frame baselines. FE refers to feature extraction, FT refers to fine-tuning while SF refers to Single-frame models. Evaluation is carried out on the WWW Crowd and violent-flow validation sets.

Model	VF: ACC	VF: AUC	WWW: mAUC	WWW: mAP
8:(5 frames, Spaced 10 Apart)	**85%**	**0.87**	0.908	0.503
8: (5 frames, Spaced 15 Apart)	84%	0.87	0.912	0.508
8: (5 frames, Spaced 20 Apart)	83%	0.86	**0.919**	**0.516**

Table 3.5: Comparing various temporal ranges covered by a late fusion 3D CNN model Carreira and Zisserman (2017) for crowd behaviour recognition. Evaluation is carried out on the WWW Crowd and violent-flows validation sets.

as the base network. Separate models are trained for each input type as well as an ensemble run where a mean prediction is taken for the two models as well as a configuration where the outputs of the penultimate layer of each network (optical flow and RGB inputs) are fused, L_1 normalisation is applied and a bank of linear SVMs are trained. Optical flow frames are generated during training/inference using the approach of Farneback (Farnebäck, 2003) with

no additional preprocessing or quantization applied. No pre-training is applied for any of the optical flow models. The results of this experiment are highlighted in table 3.6.

Model	**VF: ACC**	**VF: AUC**	**WWW: mAUC**	**WWW: MAP**
13: Late Fusion 3D Resnet OF	81%	0.84	0.890	0.430
8: Late Fusion 3D Resnet RGB	83%	0.86	0.919	0.516
14: Ensemble prediction (OF+RGB)	84%	0.86	0.926	0.553
15: SVM fused (OF+RGB)	**85%**	**0.87**	**0.929**	**0.565**

Table 3.6: Performance of optical flow and raw RGB channel inputs for crowd behaviour recognition. Evaluation is carried out on the WWW Crowd and violent-flows validation sets.

The best overall performance is achieved when fusing features from networks trained on both input types via a bank of SVMs. This method allows for the most discerning features from each network to be utilised. Optical flow features in isolation result in inferior validation scores as they only capture local motion information without any appearance features. Ideally a model which is jointly trained on optical flow and RGB frames in an early fusion manner would be developed but this is not possible with the current hardware setup without significant compromises in model capacity.

3.4.3 Comparison With The State-Of-The-Art

The best performing CNN configuration for both datasets has been determined through extensive validation. For the violent-flows collection a feature extraction approach using a pre-trained 50 layer Resnet and RGB input frames results in the best validation performance. On the other hand for the WWW Crowd set a multi-frame CNN approach which combines features from RGB and optical flow trained networks produces the best overall validation performance. This contrast in best performing approaches between the two is due to the large discrepancy

in dataset size and scene variation. With these refined configurations in place we can compare these methods to the leading techniques in the literature. A 5-fold cross validation is used to compare performance on violent-flows while the assigned test set is used for WWW crowd.

Violent-Flows Dataset

Table 3.7 compares the developed small-dataset approach with the state-of-the-art methods for the violent-flows dataset as well as the hand crafted baseline run (Hassner et al., 2012*a*). The proposed method achieves a superior mean accuracy to all of the leading techniques on the Violent-Flows dataset, highlighting the potential of deep learning methods for violent behaviour recognition.

Approach	**mACC**	**mAUC**
3: Resnet50-Feature Extraction (Proposed)	**95.2±7.5%**	**0.98**
ViF (Hassner et al., 2012*a*)	81.30±0.21%	0.85
GLCM-Texture (Lloyd et al., 2017)	86.03±4.25%	0.94
VPS (Mohammadi et al., 2016)	86.61%	N/A
MoSIFT+KDE+SC (Xu et al., 2014)	89.05±3.26%	0.9357
LaSIFT+BOW (Senst et al., 2017)	93.12±8.77%	0.97

Table 3.7: Proposed small-dataset crowd behaviour recognition approach compared to the leading techniques on the violent-flows dataset. A 5-fold cross validation is carried out in all cases, with a 95% confidence interval calculated across the 5 folds and presented alongside the mean, as is convention for this dataset.

WWW Crowd Dataset

Table 3.8 compares the developed large-dataset approach with the state-of-the-art methods for the WWW crowd test set as well as the hand crafted baseline run (Hassner et al., 2012*b*). The

proposed method achieves performance comparable to the state-of-the-art despite the limitations in hardware (specifically in terms of GPU VRAM). The inferior performance of the proposed method to the work of Shao et al. is likely due to the application of early fusion along the temporal axis in their approach, as opposed to the late fusion applied in the proposed method. More consistency is observed among class-level AUC scores, with a standard deviation of just 0.05 calculated across the 934 test videos. If we divide this standard deviation by $\sqrt{10}$ (where 10 corresponds to 10% of the overall dataset) the error for the entire dataset can be approximated as 0.0158. On the other hand, MAP scores vary a lot more with a standard deviation of 0.23 reported for the test set (0.072 for the entire dataset). No cross-validation was performed on this dataset due to the time consuming nature of experiments on a dataset of this scale. This discrepancy in standard deviation across the two metrics suggests that MAP is a much more challenging metric to score highly in for this dataset and should be focused on in future. Including additional model capacity and exposing the network to more temporal information will likely improve this performance further, as demonstrated for other video classification tasks (Carreira and Zisserman, 2017). The developed deep learning method significantly outperforms the hand crafted baseline run.

Examples of the proposed recognition model in action are presented in figure 3.6, with the 10 highest likelihood classes predicted by the network listed. A likelihood over 50% is deemed to be a trusted prediction as in many binary classification tasks. These likelihood scores correspond to the output of the final sigmoid activation for each class and are as such independent of each other. Classes correctly predicted with a likelihood over 50% are highlighted in green while classes incorrectly predicted with a likelihood above 50% are shown in red. Low likelihood predictions (below 50%) are presented in yellow and can largely be ignored while classes present in the scene but not predicted in the top 10 are shown in orange on the right hand side.

The prediction of a dancer in the beach scene is likely due to either an annotation error in the WWW Crowd training set or a particular pattern or movement resembling the dancer class. As all class likelihood scores are independent, in that they predict the presence of absence of a given class in isolation, they can contradict each other occasionally (night v. day, beach v. park). This is ultimately a flaw in the composition of the WWW crowd dataset and the recognition task put forward by the authors. The proposed model can sometimes fail to distinguish between indoor and outdoor locations, which can be quite challenging to classify with the presence of artificial lighting and highly cluttered scenes. However, classifying night and day time scenes is not directly related to crowd behaviour and is only included in this study to comprehensively evaluate the WWW Crowd dataset.

Approach	**WWW: AUC**	**WWW: mAP**
15: SVM-fused 3D CNN (Proposed)	0.931 ± 0.05	0.576 ± 0.23
7: ViF (Hand-Crafted) (Hassner et al., 2012*b*)	0.65	0.12
DLSF+DLMF (Shao et al., 2015)	0.877	0.412
3D-CNN (Ji et al., 2013)	0.86	0.39
Slicing-CNN (Shao et al., 2016)	**0.94**	**0.6255**

Table 3.8: Proposed large-dataset crowd behaviour recognition approach compared to the leading techniques on the WWW Crowd test set. Mean and standard deviation are presented for the proposed method for both metrics.

3.5 Crowd behaviour Anomaly Detection

Crowd behaviour anomaly detection methods are developed and benchmarked in this section using the LV dataset. This large-scale dataset contains a highly varied set of 28 sequences covering a range of abnormal behaviour events. Each clip contains a sequence of normal behaviour

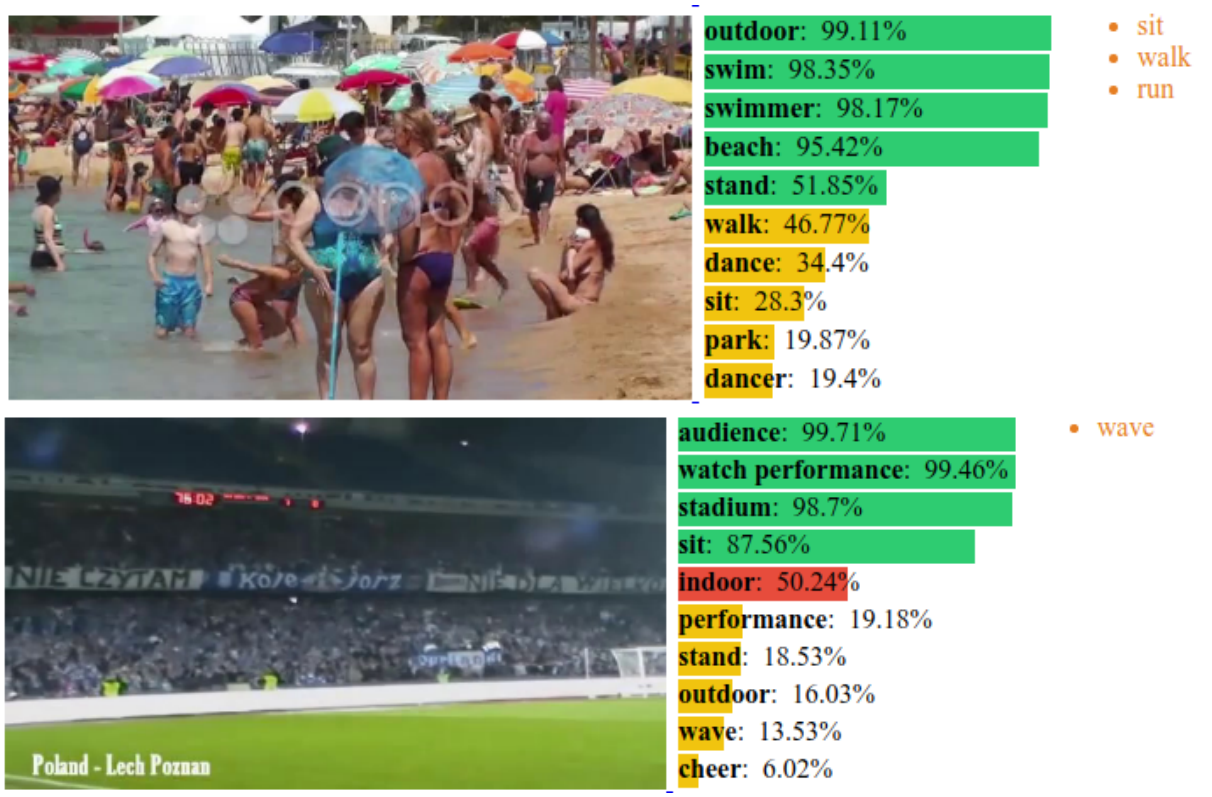

Figure 3.6: Examples of the proposed crowd behaviour recognition system in action on the WWW Crowd Dataset.

for a given camera location followed by a test section in which one or more behaviour anomalies occur. Normal behaviour frames can be used to perform any scene specific model training required by a given method. Performance is evaluated at the frame level and evaluated over an entire clip using AUC score. In total there are 68,989 abnormal frames across a total of 340,406 frames (20.5% abnormal). Validation is performed on a 5 clip subset of this collection (Crash4, Crash1, Illegal_turn, robbery3 and Kidnap) before a full comparison with the leading techniques is carried out on all clips. The following hyperparameter selection issues are optimised for this task:

- Distance metric used for outlier detection
- Single-frame v. multi-frame features
- Optical flow input v. raw RGB input v. Fusion (RGB+OF)

3.5.1 Hand-Crafted Baseline

The hand-crafted baseline run used for this task is the two-pass approach of Reddy *et al.* which utilises motion, object size and texture features to detect abnormal behaviour (Reddy et al., 2011). The scene foreground is firstly segmented, before being split into non-overlapping cells from which motion, size and texture features are extracted individually. Mean optical flow magnitude is computed for each cell and smoothed temporally to produce a simple 1-D motion magnitude feature. A 1-D object size feature is then computed within each cell by computing the segmented foreground occupancy level. Texture features are then generated using 2-D Gabor wavelets and combined with the previous features to produce a 4-D feature vector for each frame cell. Each feature is then modeled separately across all grid cells extracted from a given training set. The motion and size features are modelled using kernel density estimation while the the 2-D texture descriptor is modelled via a codebook. This codebook is generated in an online fashion where Pearson's correlation coefficient is used to measure descriptor similarity.

Once the three models are trained a two-step anomaly detection process is then applied to each grid cell in a test set. The first classifier is a simple thresholding step applied to the likelihood of a given motion feature occurring, calculated using the trained KDE model. If this first classifier deems the motion feature to be abnormal then the second classifier is called upon to further confirm this, otherwise the cell is deemed to contain normal behaviour. The second classifier analyses the likelihood of the size and texture features. If both of these fall below a certain threshold then the cell region is deemed to be anomalous. This two-step approach is used to optimise the system towards real-time computation. A spatio-temporal post-processing step is also applied to remove isolated anomalies. Any anomalous cell that does not contain at least two adjacent anomalous cells in either the temporal or spatial plane is re-classified as being normal.

3.5.2 Deep Learning Approach

For this task the proposed deep learning method uses features extracted from pre-trained crowd behaviour recognition models to perform distance based outlier detection and classify frames or frame sequences as normal or abnormal. This approach is similar in nature to many crowd behaviour anomaly detection methods albeit with the benefit of deep CNN features being used to compare samples in a more discriminative fashion. Training features are firstly extracted from the normal behaviour region within a given test clip using the activations of the penultimate layer in a given pre-trained network. A distance metric is then used at test time to compute nearest neighbour distance between a given test sample and the set of normal behaviour training sample. This distance is then thresholded to classify the observed test sample as normal or abnormal.

This type of approach analyses a given scene in a holistic fashion, extracting features from a large central crop of each frame rather than a series of smaller patches. This method requires no explicit model optimisation for a given clip, merely a feature extraction stage to generate a set of normal behaviour descriptors. The CNN models trained on the WWW Crowd dataset in the previous section are used for this task due to relevant source domain, the sheer size of the dataset and level of scene variation.

Distance Metric

The first validation experiment in this section evaluates various distance metrics for a single-frame outlier detection approach. Features are extracted from a Resnet50 model trained on the WWW Crowd dataset with an RGB input used. The hand crafted baseline approach discussed previously is also evaluated. The results of this experiment are presented in table 3.9. The cosine distance run performs best and is used for all subsequent experiments. All deep learning

approaches outperform the hand crafted baseline.

Approach	LV: AUC
6: Resnet50-Cosine distance	**0.368**
6: Resnet50-Euclidean distance	0.356
6: Resnet50-Manhattan distance	0.311
16: Hand crafted baseline (Reddy et al., 2011)	0.234

Table 3.9: Evaluation of various distance metrics for outlier detection based behaviour anomaly detection on the LV validation set.

Single-Frame v. Multi-Frame

This validation experiment compares features extracted from multi-frame and single-frame behaviour recognition networks. The pre-trained multi-frame model used (Late fusion 3D CNN) classifies each 100 frame temporal region holistically, with the overall prediction being used for all frames within this region. The results of this experiment are shown in figure 3.10

Approach	LV: AUC
8: Late Fusion 3D Resnet18 (5 frames, 20 apart)	**0.479**
6: Resnet50-Single Frame	0.368
16: Hand-Crafted Baseline (Reddy et al., 2011)	0.234

Table 3.10: Comparison of multi-frame and single-frame behaviour recognition features for outlier detection based anomaly detection. Evaluation is performed on the LV validation set.

The use of multi-frame recognition features results in significantly better AUC performance over the single-frame model despite the lack of granularity during classification (100 frames are classified at a time rather than each frame individually). This approach allows for the high-level

temporal dynamics present in the scene to be extracted and used to detect unusual behaviour.

Optical Flow v. Raw RGB v. Fusion (OF+RGB)

In the final validation experiment in this section, a comparison is made between models trained on optical flow input and those trained on raw RGB pixels, as well as an ensemble approach which combines the two feature sets. The same RGB Late Fusion 3D Resnet 18 approach as before is used for these experiments. l_1 normalisation is applied to the fused descriptor in order to account for the variation in distribution between the two feature sets. The results of this experiment are presented in figure 3.11. Isolated RGB features result in the best overall performance and are used for comparisons with the state-of-the-art. This RGB based multi-frame method captures both the visual appearance and motion dynamics of a given scene thanks to the 3D fusion step while optical flow features can only capture local motion dynamics. A fusion approach results in minor performance degradation when compared to the *RGB Late Fusion 3D Resnet18* run.

Approach	**LV: AUC**
8: RGB Late Fusion 3D Resnet18	**0.479**
13:OF Late Fusion 3D Resnet18	0.415
14: Ensemble (RGB-OF)	0.465
16: Hand crafted baseline (Reddy et al., 2011)	0.23

Table 3.11: Comparison of models trained on optical flow input, RGB input and a joint approach. Evaluation is performed on the LV validation set.

3.5.3 Comparison With The State-Of-The-Art

LV Dataset

The best performing anomaly detection configuration, found through extensive validation, is compared with the leading anomaly detection approaches on the entire LV dataset. The results of this experiment are presented in table 3.12. Examples of the proposed system in action are presented in figure 3.7.

Approach	**LV: AUC**
8: RGB Late Fusion 3D Resnet18 (Proposed)	**0.732**
16: Two-Pass Motion,Texture,Shape (Reddy et al., 2011)	0.325
Spatio-Temporal Compositions (Roshtkhari and Levine, 2013)	0.427
H.264 Features (Biswas and Babu, 2013)	0.151
150 FPS Detection (Lu et al., 2013)	0.112

Table 3.12: Comparison of the proposed anomaly detection method with the leading techniques. Evaluation is performed on the full LV dataset.

The proposed deep learning approach significantly outperforms the existing methods by a sizable margin, highlighting the value of multi-frame CNN features for behaviour anomaly detection. The developed technique utilises an existing crowd behaviour recognition model and does not require any scene specific training. The best AUC performance is achieved on scenes crash1 (0.99 AUC), panic0 (0.96 AUC) and robbery6 (0.90 AUC) while the worst performance is observed on illegal turn (0.20 AUC), kidnap (0.34 AUC) and wrong way (0.44 AUC). Inferior performance appears to occur when the abnormal event is more subtle in nature and requires a higher level understanding of the scene content and societal norms.

Figure 3.7: Examples of the proposed crowd behaviour anomaly detection system in action on the LV dataset. A single key frame from the beginning of each clip is shown. A clip level AUC of 0.98 is achieved on the first scene (crash3), which is largely static in nature until a collision happens on the road later in the sequence. However, for the second scene (fight2) a clip-level AUC of just 0.32 is achieved, due largely to the busy nature of this scene, which makes it difficult to detect the fight that occurs later on in the sequence. Clearly the level of clutter in the video sequence has an affect of detection performance. Images of anomalous events are left out of the thesis due to their potentially upsetting nature.

3.6 Discussion

Deep learning based methods have been shown to produce state-of-the-art performance in crowd behaviour recognition and behaviour anomaly detection. The *RGB Late Fusion 3D Resnet18* run results in superior performance for both tasks. This approach employs a combination of appearance and motion features captured over a wide temporal region (100+ frames). Training such a model does however require large quantities of training data (100,000+ samples). Single-frame feature extraction performs better for small dataset behaviour recognition problems where the number of discrete training samples needs to be maximised. The inferior performance of the proposed method to the work of Shao et al. is likely due to the application of early fusion along the temporal axis in their approach, as opposed to the late fusion applied in the proposed method.

The ability to re-purpose behaviour recognition models for behaviour anomaly detection

provides additional functionality to these models. Developing this type of approach enables a supervised learning model to be trained for a known set of behaviour concepts and then deployed in an unsupervised manner to detect any unknown/abnormal behaviour concepts.

Training video analysis models using deep learning methods requires significant computational resources to achieve high levels of predictive performance. This computational bottleneck limits the possible performance of the methods developed using the current hardware setup. Despite these constraints, strong performance can been achieved using workarounds such as lower capacity models and spacing out the frames ingested into a late fusion CNN model rather than processing all frames in a region.

3.7 Summary

In this chapter deep learning based approaches to crowd behaviour recognition and behaviour anomaly detection are developed. Various model selection issues including network capacity, length of the temporal region observed and data preprocessing steps are explored, leading to high predictive performance in each analysis task. State-of-the-art performance is achieved on the violent-flows and LV datasets while competitive performance is achieved on the WWW Crowd dataset. The superiority of deep learning approaches to hand crafted features for crowd behaviour analysis is demonstrated throughout this chapter.

Chapter 4

Crowd Congestion Analysis Via Deep Neural Networks

4.1 Introduction

This chapter investigates the use of deep neural network techniques for the computer vision tasks of crowd counting and crowd density level estimation. Research question 1 is addressed in this chapter as well as in chapter 3. Various convolutional neural network configurations, preprocesing steps and implementation strategies are evaluated for each task in an attempt to find the best practices for deep learning based crowd congestion analysis. As in chapter 3 a baseline run using hand crafted features is included for each task to compare performance with deep learning methods. Following the development of a refined method for each task using a validation set, comparisons are made with the leading techniques from the literature using a larger test set. The work in this chapter was published at VISAPP 2017 under the title "Fully convolutional crowd counting on highly congested scenes" (Marsden et al., 2016).

4.2 Contributions

The main contributions of this chapter are listed below:

- A patch-based regression approach to crowd counting is developed;
- A crowd density level estimation dataset is constructed;
- The proposed technique is shown to be superior to a hand crafted baseline for both crowd counting and crowd density level estimation;
- State-of-the-art performance is achieved on the ShanghaiTech crowd counting dataset.

4.3 Experimental Framework

The proposed crowd congestion analysis models are developed using a common framework in which certain hyperparamaters and model selection choices are kept consistent across all experiments. These limit the parameter space to explore during validation and focus experimentation on the more impactful model selection issues.

4.3.1 Fixed Hyperparameters

The majority of the fixed parameters for this set of experiments are kept consistent with those used in the previous chapter. The common framework used for all experiments in this chapter is listed in table 4.1. The only changes to the previous chapter are the objective functions used (detailed in full below) and the absence of temporal data augmentation due to the single frame nature of the datasets used.

Optimiser	Adagrad (Duchi et al., 2011)
CNN Architecture	Resnet (18, 50 layers)
Regularisation	L_2 Weight Decay (0.001)
Augmentation	Random Crops, Random Flips
Initalisation	(Glorot and Bengio, 2010), bias terms set to 0
Loss Function	MSE/BCE for count regression, CCE/MSE for DL Estimation
Hardware	4GB Nvidia GTX 970 GPU, 8 core Intel i7 CPU, 32GB RAM

Table 4.1: Common framework used for all crowd congestion analysis training runs

Loss Functions

For regression-based crowd counting problems a mean squared error loss, given in equation 4.1, is minimised. $\hat{S}_i$ refers to the predicted value while S_i is the corresponding ground truth values.

$$L_{\text{MSE}} = \sum_{i=1}^{K} (S_i - \hat{S}_i)^2 \tag{4.1}$$

Two types of loss function are investigated for heatmap generation based crowd counting. First, a pixel-wise mean squared error is employed (in this case the MSE loss is summed for all pixels in the image). Second, a combination of a binary cross entropy loss and a sigmoid activation on the final output layer is investigated. Binary cross entropy loss is presented again in 4.2.

$$L_{\text{BCE}} = -\sum_{j=1}^{K} S_j \log(\hat{S}_j) + (1 - S_j) \log(1 - \hat{S}_j) \tag{4.2}$$

For crowd density level estimation two different loss functions are evaluated. First, for classification based density level estimation a categorical cross entropy loss, given in equation 4.3, is minimised following a softmax activation on the final network output. Second, a regression

based density level estimation approach is investigated with a mean squared error minimised.

$$L_{\text{CCE}} = -\sum_{j=1}^{K} S_j \log(\hat{S}_j) \tag{4.3}$$

4.3.2 Model Selection Issues Investigated

With this framework in place the experimental focus of this chapter can be discussed. The following model selection issues are investigated for the task of crowd counting:

- Patch regression counting v. heatmap generation counting;
- Patch size used for training and inference of crowd counting models;
- Model capacity (number of network layers);
- Trainable parameters: training from scratch v. fine-tuning v. feature extraction.

The following model selection issues are investigated for crowd density level estimation:

- Model capacity (number of network layers);
- Trainable parameters: training from scratch v. fine-tuning v. feature extraction;
- Classification based DLE v. regression based DLE.

4.4 Crowd Counting

Crowd counting methods are developed and benchmarked in this section using the ShanghaiTech dataset (parts A and B evaluated separately). These two datasets represent a low-medium congestion and medium-high congestion crowd counting dataset respectively. Both

datasets are evaluated at the frame level, with dot map annotations highlighting the head location of each person included for all images. Performance is evaluated using mean absolute error (MAE) and root mean squared error (MSE). For model selection experiments a 50 image validation set is taken from the training set of parts A and B. Training is carried out for 10,000 iterations for all experiments. The number of training set epochs this corresponds to varies with the dataset used and the model trained (which in turn affects the maximum batch size). The full details of each dataset is listed in table 4.2. The following model selection issues are investigated for this task:

- Patch regression counting v. heatmap generation counting;
- Patch size used for training and inference of crowd counting models;
- Model capacity (number of network layers);
- Trainable parameters: training from scratch v. fine-tuning v. feature extraction.

Dataset	**Count Range**	**Training Set Size**	**Validation Set Size**	**Test Set Size**
ShanghaiTech A	33-3139	300 Frames	50 Frames	182 Frames
ShanghaiTech B	5-600	400 Frames	50 Frames	316 Frames

Table 4.2: Training, validation and test set sizes for the ShanghaiTech Dataset. 50 frames are removed from each training set to form a validation set, reducing the training set size for any model selection experiments.

4.4.1 Hand-Crafted Baseline

The hand crafted feature baseline used for this experiment is the support vector regression approach of Idress *et al.* (Idrees et al., 2013). This method combines HOG-based head detection (Dalal and Triggs, 2005) with texture features generated using the SIFT descriptor (Lowe, 2004)

and frequency-domain features produced via the Fourier transform to perform count regression. A global consistency constraint is then employed using a Markov Random field, catering for local disparities in count estimates.

4.4.2 Deep Learning Approach

Patch-based Regression v. Heatmap Generation

The first validation experiment for crowd counting compares patch-based count regression with heatmap generation based counting on the ShanghaiTech validation sets. Patch based regression is simply regression based counting where the image is processed in sections due to hardware limitations. Patch based count regression divides a given frame into a set of non-overlapping patches, passing each patch through a CNN regressor before accumulating the overall count for the scene. A given CNN architecture (e.g. Resnet18) can be converted to a regressor by updating the final layer to produce a single output value and applying a ReLU activation afterwards. Training is carried out at the patch level, enabling large amounts of training samples to be produced from a given training image, particularly when random cropping is employed.

Heatmap based counting on the other hand produces a pixel-wise map of crowd congestion which when integrated produces an estimate of the overall crowd count. Any CNN architecture (e.g. Resnet18) can be converted to perform pixel-wise heatmap generation by taking the output of a given convolutional layer in the network and adding a single kernel convolutional layer and any necessary upscaling required to match the input image. Billinear upscaling is applied for all experiments. Training is performed using pairs of image patches and pre-generated heatmap ground truth images (Zhang et al., 2016). These ground truth heatmap images are generated by setting the pixel location of each person to 1.0 and applying guassian bluring to the whole ground truth patch. Heatmap based counting via a CNN is illustrated in figure 4.1. Gaussian

blurring via a 3×3 kernel is applied to all ground truth heatmaps for training. This blurring eases the learning task for heatmap generation. Both mean squared error (MSE) and binary cross entropy (BCE) are investigated as loss functions for heatmap generation. BCE loss looks at each pixel individually and considers whether a person is present or not while MSE looks at the overall heatmap error.

The Resnet18 architecture is used for all runs in this experiment with the entire network trained end to end and initialised from an ImageNet pretrained model. 100×100 images patches are used for training all runs in this case. For heatmap based counting models the output of the 9th convolutional layer is used to produce the estimated heatmap as going beyond this layer reduces the feature maps to 1/8 the input size, while the 9th layer output is 1/4 the original size (allowing easier upscaling). The results of this experiment are presented in table 4.3. Patch-based regression far exceeds the performance of either heatmap based approach as well as the hand-crafted baseline. This is due to the inability of the heatmap-based approach to handle low-density crowds as this approach treats a large crowd as a texture or pattern rather than a collection of discrete objects. The use of BCE loss results in superior performance to MSE loss for heatmap generation counting, possibly due to the sigmoid activation function limiting the range of the count prediction values, resulting in a more robust estimator. Patch regression based counting will form the basis of all subsequent validation experiments.

Model Capacity, Trainable Parameters

The next validation experiment compares regression based counting networks of various depths (Resnet18 and Resnet50) while also comparing training strategies for each (training from scratch, fine-tuning and feature extraction). Training is carried out for 10,000 iterations for all runs with a patch size of 100×100 used during training and inference. The results of this experiment are

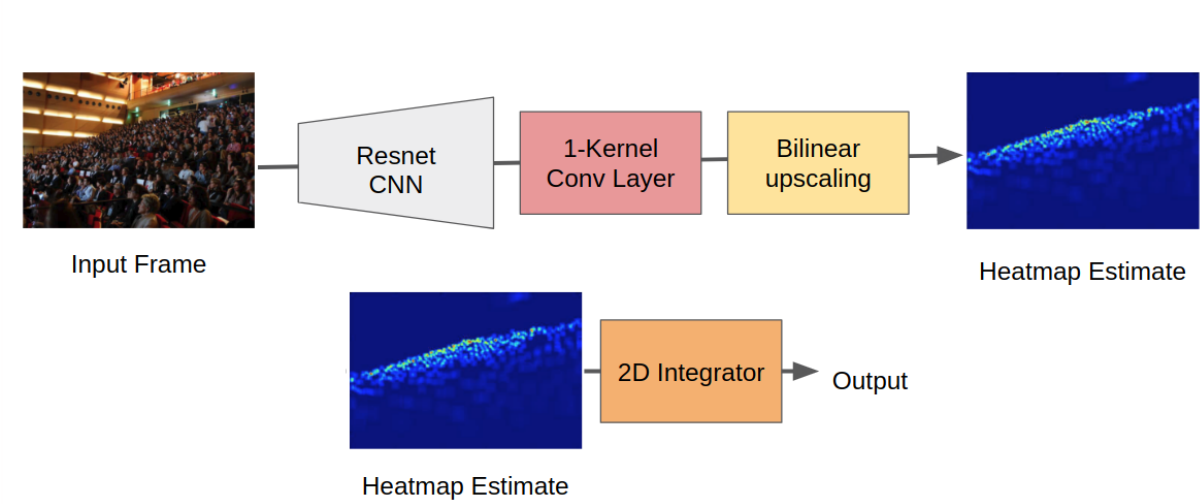

Figure 4.1: Heatmap based crowd counting via CNN. A network is trained to estimate congestion heatmaps using a set of ground truth images. An estimated heatmap is then integrated to produce an estimate of the overall crowd count.

Model	**Part A: MAE**	**Part A: MSE**	**Part B: MAE**	**Part B: MSE**
Resnet18-Regressor	**133.8**	**245.74**	**9.23**	**13.42**
Resnet18-Heatmap-MSE	229.23	391.02	80.08	99.98
Resnet18-Heatmap-BCE	177.4	278.6	34.3	49.18
(Idrees et al., 2013)	256.3	410.5	86.5	102.3

Table 4.3: A comparison of patch regression and heatmap generation based crowd counting on the ShanghaiTech validation sets (parts A and B). The hand-crafted baseline approach of Idress *et al.* (Idrees et al., 2013) is also evaluated.

presented in table 4.4. The best overall performance is achieved when a Resnet18 model is fine-tuned from a pre-trained ImageNet model. Including additional model capacity via Resnet50 results in inferior performance most likely due to overfitting. In all cases transfer learning from a pre-trained model resulted in superior performance and the deep learning runs outperform the hand crafted baseline.

Architecture	Config	Part A: MAE	Part A: MSE	Part B:MAE	Part B:MSE
Resnet18	FS	145.6	275.3	11.3	14.3
Resnet18	FT	**133.8**	**245.74**	**9.23**	**13.42**
Resnet18	FE	160.2	295.2	17.8	27.81
Resnet50	FS	190.6	310.9	16.5	22.5
Resnet50	FT	177.5	303.5	13.6	19.6
Resnet50	FE	145.9	263.8	12.83	18.26
Hand-Crafted	N/A	256.3	410.5	86.5	102.3

Table 4.4: Comparison of the various network architectures and training strategies for patch regression based crowd counting on the ShanghaiTech validation sets (Part A and B). FS refers to From Scratch, FT refers to Fine Tuning while FE refers to Feature extraction.

Patch Size During Training and Inference

The final model selection experiment for crowd counting compares various patch sizes used for CNN based count regression. A larger patch size will expose the network to more scene context but will result in a less homogeneous object size (i.e. people in the scene) due to camera perspective issues. A smaller patch size on the other hand will result in a more homogenous object size but the inclusion of less scene context. Patches of size 50×50, 100×100 and 200×200 are evaluated. Training is carried out using a pre-trained Resnet18 for 10,000 iterations in each case. The results of this experiment is shown in figure 4.5

A 100×100 patch size during model training and inference results in the best overall validation performance. This size results in the best tradeoff between scene context and variation in object size. Significantly inferior performance is observed for 200×200 patches, highlighting the challenge in training a counting model for images containing objects of significantly

Patch Size	Part A: MAE	Part A: MSE	Part B:MAE	Part B:MSE
50×50	142.1	259.3	10.3	14.4
100×100	**133.8**	**245.74**	**9.23**	**13.42**
200×200	190.5	287.5	19.5	28.3
(Idrees et al., 2013)	256.3	410.5	86.5	102.3

Table 4.5: Comparison of the various patch sizes used for regression based crowd counting on the ShanghaiTech validation sets (Part A and B).

different sizes.

4.4.3 Comparison With The State-Of-The-Art

The best performing crowd counting configuration on the ShanghaiTech validation sets involves a Resnet18 patch-based regressor trained end-to-end on 100×100 patches and initialised from a pre-trained ImageNet model. This configuration is compared with the leading techniques in the literature as well as a hand-crafted baseline on the test sets of the ShanghaiTech dataset (parts A and B). Table 4.6 presents the results of this experiment. The proposed technique achieves state-of-the-art performance on the ShanghaiTech part B test set and competitive performance on the ShanghaiTech part A test set. However when the two datasets are considered together the proposed technique can be deemed superior to the work of Sindagi and Patel as it achieves a 48% better MAE score on part B and just a 13% inferior MAE score on part A. This line of thinking assumes that both datasets are of equal importance and that accuracy in low congestion scenes is as important as in high congestion scenes. The proposed technique also employs fewer trainable model parameters, with a single resnet18 model trained compared to the 3 branch network of Sindagi and Patel. Examples of the proposed counting model in action are presented in figure 4.2. Larger errors are observed for higher congestion scenes, due to the limited number

of pixels occupied by each person.

Approach	Part A: MAE	Part A: MSE	Part B:MAE	Part B:MSE
(Idrees et al., 2013)	160.5	225.6	65.4	89.7
(Zhang et al., 2016)	110.2	173.2	26.4	41.3
(Sam et al., 2017)	90.4	135.0	21.6	33.12
(Sindagi and Patel, 2017)	**73.6**	**106.4**	20.1	30.1
Patch-Regressor (Proposed)	83.62	131.5	**12.61**	**23.6**

Table 4.6: Comparison of the leading CNN based crowd counting approaches on the ShanghaiTech test sets (parts A and B). Not all methods listed here are included in the literature review as many of the approaches apply a very similar heatmap generation approach.

4.5 Crowd Density Level Estimation

Crowd density level estimation (DLE) methods are developed and benchmarked in this section using a re-purposed version of the ShanghaiTech dataset that combines parts A and B in order to maximise the range of crowd congestion level. This new dataset, which will be referred to as *ShanghaiTech Density*, is proposed here since the existing collections for this task contain only low congestion scenes, with no dataset commonly accepted by the research community. Performance is evaluated for density level estimation at the frame level using mean absolute error, accuracy, and top-2 accuracy. For these experiments the entire frame is processed in a single CNN forward pass, with appropriate downsampling and cropping applied. Model selection experiments are carried out on a 100 image validation subset taken from the combined training sets of parts A and B. Training is carried out for 10,000 iterations for all experiments. The following model selection issues are investigated for this task:

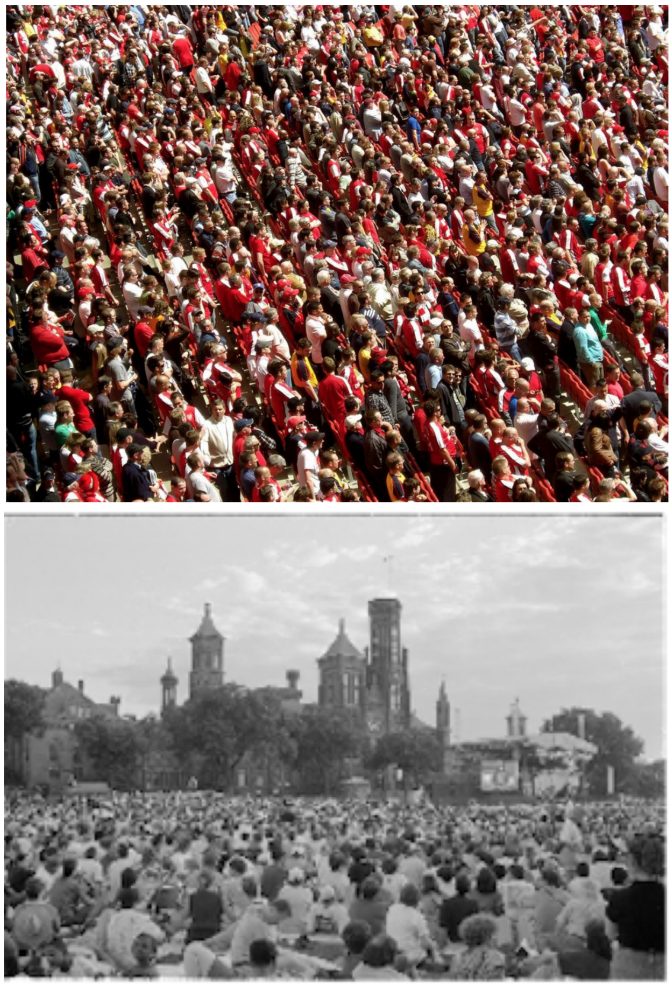

Figure 4.2: Examples of the proposed crowd counting system in action on the ShanghaiTech dataset. The first image contains 803 people with an estimated count of 819 calculated. The second image contains 1544 people with an estimated count of 1394 calculated. Larger errors are observed for higher congestion scenes.

- Classification based density level estimation v. regression based density level estimation;
- Model capacity (number of network layers);
- Trainable parameters: training from scratch v. fine-tuning v. feature extraction.

4.5.1 ShanghaiTech Density Dataset Construction

The Shanghaitech dataset is combined into a single collection by joining the training, validation and test sets of ShanghaiTech parts A and B. The overall breakdown of this new Shanghaitech

Density dataset is presented in table 4.7. This provides a significant increase in crowd density variation compared to using either set individually. The overall crowd count value associated with each image in the original dataset is then used to infer a crowd density level label. The annotation scheme presented in table 4.8 is used to this end. This scheme provides greater granularity for low congestion scenes which are more commonly encountered in public. Crowds of up to 5000 people are covered by this scheme, providing far greater range than the existing datasets and schemes.

Dataset	**Training Set Size**	**Validation Set Size**	**Test Set Size**
ShanghaiTech Density	700 Frames	100 Frames	498 Frames

Table 4.7: Training, validation and test set sizes for the ShanghaiTech Density Dataset.

Density Level Label	**Min Count**	**Max Count.**
0	0	49
1	50	99
2	100	149
3	150	249
4	250	499
5	500	999
6	1000	1999
7	2000	2999
8	3000	3999
9	4000	5000

Table 4.8: Annoation scheme used for the ShanghaiTech Density dataset

The distribution of the ShanghaiTech Density dataset across these 10 density levels is illus-

trated in figure 4.3. The distribution of an evenly spaced annotation scheme (0,500,1000,1500 etc) is shown in figure 4.4. This scheme results in an extremely skewed distribution with the majority of the samples falling under label 0. Therefore the annotation scheme which provides more granularity to lower congestion scenes is employed.

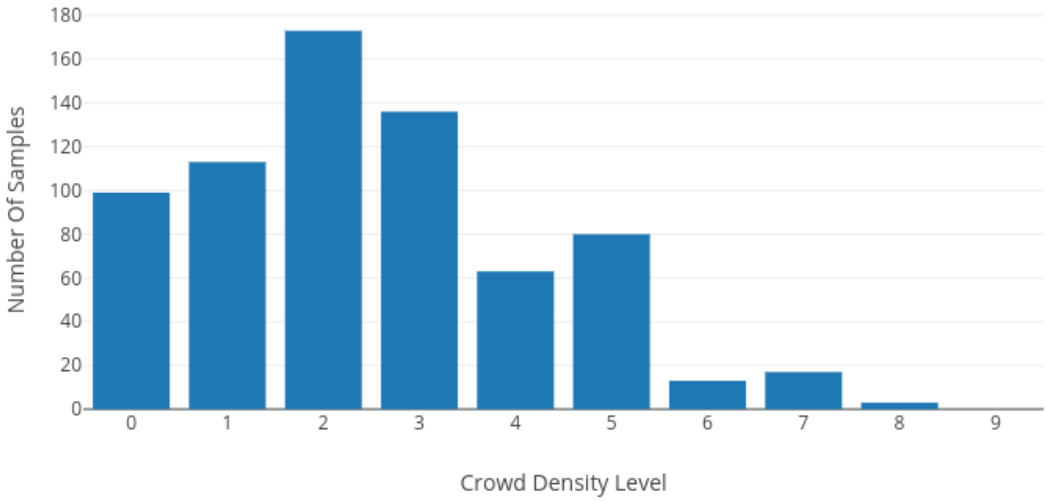

Figure 4.3: Distribution of the ShanghaiTech Density dataset across the 10 density level labels using the proposed annotation scheme.

4.5.2 Hand-Crafted Baseline

The hand crafted baseline for this experiment uses SIFT descriptors (Lowe, 2004) to generate a bag of visual words model. 128-D SIFT descriptors are extracted in a dense manner from all training images. These descriptors are then clustered into 512 codewords using k-means clustering. The normalised distribution of SIFT descriptor occurrence across these 512 codewords is then used as a fixed length descriptor for comparing images. 10 one-v-rest support vector machines are then trained to classify crowd density images using the proposed fixed length descriptor. This overall pipeline is then used to classify the crowd density level of a given image.

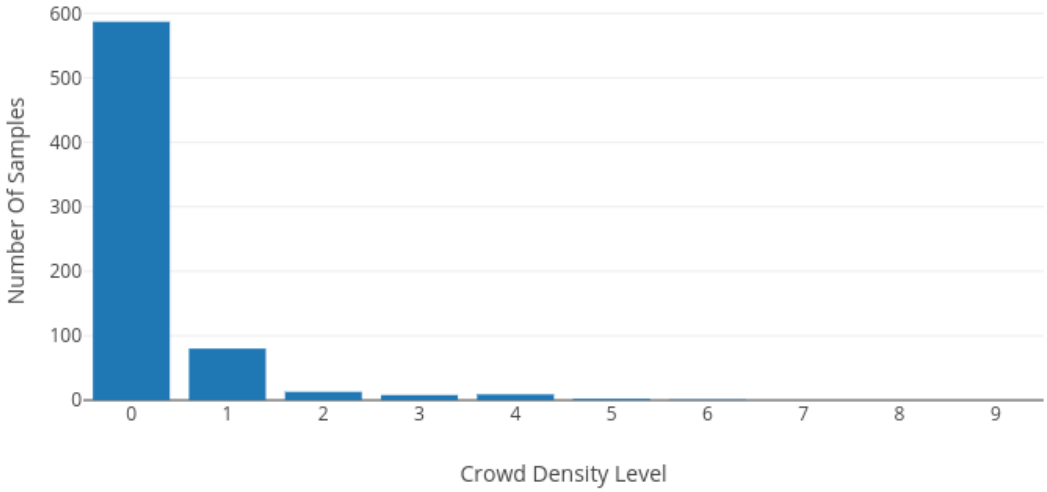

Figure 4.4: Distribution of the ShanghaiTech Density dataset across the 10 density level labels using an evenly spaced annotation scheme. This results in an extremely skewed distribution across the 10 density levels. It is reminiscent of the Zipf Parento distribution (Powers, 1998)

4.5.3 Deep Learning Approach

Classification based Crowd DLE Vs. Regression based Crowd DLE

The first validation experiment in this section compares classification based Crowd density level estimation (DLE) with regression based Crowd DLE. For classification models, the final network layer will contain 10 neurons and be followed by a softmax activation. For regression models, the final network layer will contain a single neuron and be followed by a ReLU activation. Regression outputs are rounded to the nearest whole number to produce a density label for evaluation. The Resnet18 architecture will be trained end-to end for each run after being initialised from a pre-trained ImageNet model. The results of this experiment are presented in table 4.9. Classification-based DLE achieves the best overall performance despite the ordinal relationships between class labels. Both deep learning runs comfortably outperform the hand

crafted baseline. Classification based estimation is used for all subsequent experiments in this chapter.

Approach	MAE	Accuracy	Top-2 Accuracy
Classification-based DLE	**0.62**	**52%**	**89%**
Regression-based DLE	0.71	50%	87%
Dense SIFT+ BOW (Hand Crafted)	1.14	33%	72%

Table 4.9: Comparison of regression-based DLE, classification-based DLE and the hand crafted baseline run. Evaluation is carried out on the ShanghaiTech Density validation set.

Model Capacity, Training Strategy

The next experiment compares classification based DLE models of various depths (Resnet18 and Resnet50) while also comparing training strategies for each (training from scratch, fine-tuning and feature extraction). The results of these experiments are presented in table 4.10. A fine-tuned Resnet18 model performs best for MAE score and jointly best for accuracy, while a Resnet50 feature-extractor performs best for top-2 accuracy. Due to the ordinal nature of this problem more importance is placed on MAE score and therefore the Resnet18 fine tuned run is deemed to be the best performing configuration and is used for all subsequent experiments. In all cases transfer learning from a pre-trained model resulted in superior performance.

4.5.4 Comparison With The State-Of-The-Art

The proposed technique is evaluated on the ShanghaiTech Density test set and compared with the hand crafted baseline as well as the previously developed crowd counting model being re-purposed for the DLE task. Crowd counting estimates are quantized into a crowd density level label using the previously proposed annotation scheme. The results of this experiment

Approach	MAE	Accuracy	Top-2 Accuracy
Resnet18-Scratch	0.67	44%	0.82%
Resnet18-FE	0.90	40%	81%
Resnet18-FT	**0.62**	**52%**	89%
Resnet50-Scratch	1.02	29%	62%
Resnet50-FE	0.66	52%	**91%**
Resnet50-FT	0.95	32%	68%
Dense SIFT+ BOW (Hand Crafted)	1.14	33%	72%

Table 4.10: Comparison of various model depths and training strategies for classification based density level estimation on the ShanghaiTech Density validation set.

are presented in table 4.11. The best overall performance is achieved when a crowd counting model is re-purposed for density level estimation. This performance boost, however, results in significantly more computational demand due to the patch-based regression method used, with each patch individually processed through the network. Both methods significantly outperform the hand crafted baseline run. The method used for a given application ultimately depends on the available computational resources, the need for real-time processing, and the minimum margin of error required.

Approach	MAE	Accuracy	Top-2 Accuracy
Resnet18-FT	0.65	47.7%	89.9%
Resnet18-Count-Quantized	**0.28**	**72.4%**	**99.5%**
Dense SIFT+ BOW (Hand Crafted)	0.95	32.7%	80.1%

Table 4.11: Comparison of various DLE techniques on the ShanghaiTech Density test set.

4.6 Discussion

Deep learning based methods have been shown to produce state-of-the-art performance in crowd counting and very strong density level estimation performance on a newly proposed benchmark. The lack of training data available for both tasks is overcome through aggressive data augmentation (i.e. random cropping and flipping during optimisation). With the availability of larger and more varied datasets the performance of these data-driven models should only improve further.

The best performing crowd counting run employs patch-based CNN regression trained end-to-end on 100×100 patches. The superiority of patch regression counting to heatmap based counting is due to the inability of heatmap based counting to deal with low congestion scenes where the crowd is a collection of individual object/people. Heatmap counting treats the overall crowd as a texture or pattern and is therefore suited mainly to high congestion scenes. Patch regression can however scale to both settings. This is a computationally expensive method but leads to highly accurate counting models.

The best performing DLE run applies Softmax classification on top of a lower capacity network due to the lack of available training data. Density level estimation can be considered a coarse, surface level approximation of crowd counting. The developed DLE method, which processes an entire frame in a single CNN forward pass is far less computationally demanding than the proposed counting method and requires only a single label per frame rather than a dot annotation for each individual object/person. The choice of which method to use ultimately depends on the hardware resources available, whether real-time processing is needed, and the acceptable margin of error when measuring the congestion of a crowded scene via CCTV footage.

4.7 Summary

In this chapter deep learning based approaches to crowd counting and crowd density level estimation are developed and compared to the leading approaches from the literature. Various model selection issues including model capacity, choice of loss function and training strategy are explored for both tasks. State-of-the-art performance is achieved on the ShanghaiTech dataset for crowd counting while promising initial performance is achieved on the newly proposed ShanghaiTech Density dataset. The superiority of deep learning approaches to handcrafted features for crowd congestion analysis is once again demonstrated throughout this chapter.

Chapter 5

Multi-Task Crowd Analysis

5.1 Introduction

This chapter investigates the use of multi-task learning (MTL) techniques for deep neural network based crowd video analysis. Research question 2 is addressed by this chapter. Auxiliary loss terms are first investigated as a means to improve the performance of a given crowd analysis task without introducing any additional data or annotation labels. Joint training of complementary crowd analysis tasks (behaviour recognition, crowd counting, density level estimation) is then investigated to see if any improvements in predictive performance can be achieved while reducing the overall network parameter count across multiple tasks. Various model selection issues are refined for each multi-objective experiment using the relevant validation sets in each case. A multi-task crowd analysis model is then compared with the already proposed single objective approaches to each task as well as the leading approaches in the literature. The work in this chapter was published at AVSS 2017 under the title "ResnetCrowd: A Residual Deep Learning Architecture for Crowd Counting, Violent Behaviour Detection and Crowd Density Level Classification" (Marsden et al., 2017).

5.2 Contributions

The main contributions of this chapter are listed below:

- Auxiliary loss functions are shown to improve crowd density level estimation performance with a negligible increase to the overall parameter count;
- A joint model for crowd counting and behaviour recognition is developed, improving the predictive performance of both tasks and reducing the overall parameter count by 50%.

5.3 Experimental Framework

As in the previous chapters, the proposed multi-task crowd analysis models are developed using a common framework in which certain hyperparamaters and model selection choices are kept consistent across all experiments. This limits the parameter space to explore during validation and focus experimentation on the more impactful model selection issues.

5.3.1 Fixed Hyperparameters

All of the fixed parameters for this set of experiments are kept consistent with those used in chapters 3 and 4. The common framework used for all experiments in this chapter is listed in table 6.1.

5.3.2 Model Selection Issues Investigated

The following model selection issues are investigated for auxiliary loss experiments:

- Loss term weightings (equal v. manual v. data-driven).

The following model selection issues are investigated for joint task training experiments:

Optimiser	Adagrad (Duchi et al., 2011)
CNN Architecture	Resnet (18, 50 layers)
Regularisation	L_2 Weight Decay (0.001)
Augmentation	Random Crops, Random Flips
Initalisation	(Glorot and Bengio, 2010), bias terms set to 0
Loss Functions	MSE, BCE, CCE
Hardware	4GB Nvidia GTX 970 GPU, 8 core Intel i7 CPU, 32GB RAM

Table 5.1: Common framework used for all multi-task analysis training runs.

- Task specific normalisation v. shared normalisation;
- Loss term weightings (equal v. manual v. data-driven);
- 2-task training v. 3-task training.

5.4 Auxiliary Loss Functions

Auxiliary loss terms can be included in the objective function employed when optimising a neural network to add additional regularisation to the model or induce a bias towards learning a certain type of function by penalising certain conditions. An example of this is shown in equation 5.1 where two separate loss terms and a regularisation term make up the overall objective function. This additional loss term can be calculated using the existing training data X or it may require a separate ground truth to be generated from the existing labels. After training is completed, a given neural network can be used for inference exactly as before, even if auxiliary network outputs are included.

$$C(X;W) = L_1(X;W) + L_2(X;W) + R(W) \tag{5.1}$$

Weightings can be applied to the various loss terms within a given objective function to give a greater influence to a certain loss term or terms during optimisation as shown in equation 5.1. These weightings can either be set manually, inferred in a data driven way, or set to 1.0 resulting in an equal weighting for all terms.

Crowd counting and density level estimation are both well suited for research into auxiliary loss terms, given that both tasks have already been attempted using various loss functions in chapter 4. Both of these tasks are investigated for any potential performance boosts associated with an auxiliary loss training strategy. The following model selection issues are investigated for auxiliary loss experiments:

- Loss term weightings (equal v. manual v. data-driven).

5.4.1 Density Level Estimation

CNN based crowd DLE is investigated in the previous chapter (sections 4.5) using both mean squared error (MSE) loss for regression based estimation and categorical cross entropy (CCE) loss for classification based estimation. While classification based estimation performed better of the two, a regression based auxiliary loss may improve performance as it recognises the ordinal relationship between density level classes unlike classification based estimation. This auxiliary regression output can be included through an additional fully connected output layer with a single neuron. This concept is visualised in figure 5.1. These two network outputs are then used to compute the two loss terms within the overall objective function for this task. This new approach can then be trained in an identical manner to the original single task DLE method with the classification output serving as the *primary* output used for inference and evaluation.

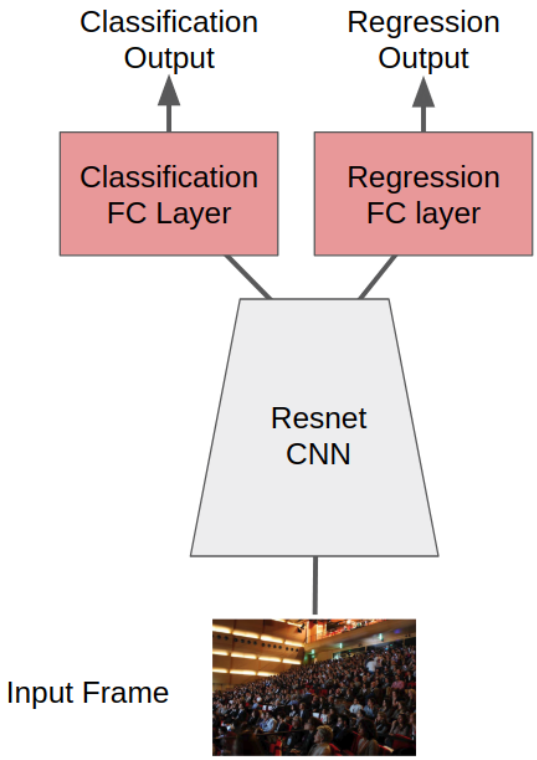

Figure 5.1: Auxiliary regression loss output included for crowd density level estimation

Loss Weightings

Various loss weighting schemes are investigated for auxiliary loss training on the ShanghaiTech Density validation set. These include several fixed weighting schemes as well a data driven scheme that attempts to balance the influence of all loss terms on weight updates. The proposed data-driven scheme produces a set of loss weights based on the gradient magnitude calculated for each network output over the entire training set. Network weights are frozen while these gradient magnitudes are computed, before any optimisation steps have been taken. These output specific gradient values correspond to the influence each output has upon weight updates during optimisation. Once these gradient values are computed, the weighting for each loss term W_i is computed using equation 5.2. Outputs with a higher gradient receive a lower weighting using this scheme in an attempt to balance the influence of various loss terms during optimisation. This set of loss weights are then applied for the entire optimisation process.

$$W_i = \frac{\max_i \nabla L_i(x; W)}{\nabla L_i(x; W)} \tag{5.2}$$

The Resnet18 architecture is trained end to end for all training runs while being initialised from a pre-trained imageNet model. The results of this experiment are presented in table 5.2. W_1 refers to the weighting for cross entropy loss (classification) while W_2 refers to the weighting for MSE loss (regression). Equal loss weightings achieves the best overall validation performance as MAE performance is deemed to be more important than accuracy due to the ordinal relationship between classes. Data driven loss weighting does not have a significant influence in this experiment. This may be due to the lack of variation in gradient magnitudes between network outputs in this case.

Model	W_1	W_2	**MAE**	**ACC**	**Top-2 ACC**
Resnet18	1.0	0.5	0.64	52%	90%
Resnet18	0.5	1.0	0.63	53%	90%
Resnet18	1.0	1.0	**0.62**	**54%**	**90%**
Resnet18	Data Driven	Data Driven	0.63	53%	**90%**

Table 5.2: Density level estimation performance for various weighting schemes on the ShanghaiTech Density validation set.

Comparison To The State-Of-The-Art

The best performing auxiliary loss configuration for crowd DLE is compared to the other approaches developed in chapter 4 on the ShanghaiTech Density test set, the results of which are presented in table 5.3. The inclusion of the auxiliary regression loss improves performance over the single loss baseline for all metrics with a negligible increase in network parameters and inference time. This approach is still however inferior in terms of benchmarking performance to

the computationally heavy patch-based counting method, but closes the gap while maintaining the low computational cost.

Approach	MAE	Accuracy	Top-2 Accuracy
Resnet18-FT	0.65	47.7%	89.9%
Resnet18-FT-Auxiliary loss	0.644	49.1%	90.1%
Resnet18-Count-Quantized	**0.28**	**72.4%**	**99.5%**
Hand Crafted Baseline	0.95	32.7%	80.1%

Table 5.3: Comparison of various DLE techniques on the ShanghaiTech Density test set including the hand-crafted baseline run and the quantized crowd count method proposed in chapter 4.

5.4.2 Crowd Counting

CNN based crowd counting is investigated in chapter 4 (section 4.4) as both a patch-based regression model and a heatmap-generation model, optimised using a patch level MSE loss and pixel level MSE loss respectively. A patch-level MSE lacks any local spatial context which can be provided by the pixel level MSE of heatmap based counting. Training a patch-based regression model which includes a heatmap generation auxiliary loss may add some robustness to the count regressor. The Shanghaitech dataset (parts A and B) are again used to evaluate validation performance. The Resnet18 architecture is used for this experiment, with the patch-based regression and heatmap generation counting architectures used in chapter 4 combined into a single model. The heatmap generating convolutional layer is added onto the 9th convolutional layer of Resnet18 as before while the regression output layer is included at the very top of the network. This architecture is visualised in figure 5.2. Training is carried out end-to-end in a single run, with heatmap ground truth images and corresponding count values generated to calculate the two loss terms.

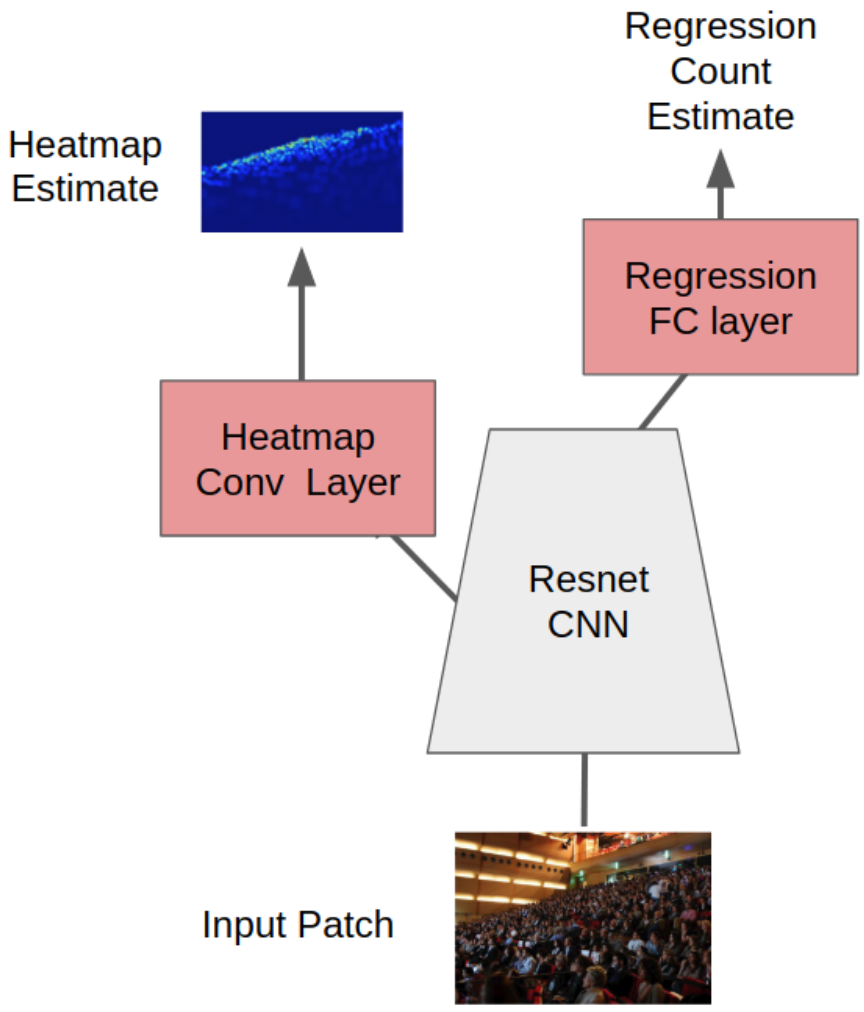

Figure 5.2: Auxiliary heatmap generation output included for patch based crowd counting

Loss Weightings

Various weightings schemes are investigated for auxiliary loss based crowd counting including the data driven scheme discussed in the previous section. The results of this experiment are presented in table 5.4. The data-driven weighting scheme results in the best overall performance, while there is little variation between the various fixed weighting schemes. The superiority of the data-driven scheme in this case is likely due to a significant difference in the gradient magnitudes of pixel level and patch-level output layers.

Heatmap Weight	Reg. Weight	PART A: MAE/MSE	PART B: MAE/MSE
0.5	1.0	135.6/231.4	13.1/16.7
1.0	0.5	136.12/213.4	12.1/16.3
1.0	1.0	136.12/**213.4**	12.1/16.3
Data-Driven	Data-Driven	**130.1**/228.6	**9.6/13.3**

Table 5.4: Crowd counting performance for various auxiliary loss weighting schemes on the ShanghaiTech validation sets (Part A and B).

Comparison With The State-of-the-Art

The optimal auxiliary loss configuration for patch-based crowd counting is compared to the single-task approach developed in chapter 4 as well as the leading approaches from the literature. Performance is evaluated on the ShanghaiTech test set (part A and B), the results of which are presented in table 5.3. The inclusion of the auxiliary loss term reduces MAE for Part A by 2% but at the cost of an increase in MAE on part B by 4.7%. The heatmap loss appears to be more suited to higher congestion scenes and improves MSE performance for both parts of the dataset.

Approach	PART A: MAE/MSE	PART B: MAE/MSE
(Sam et al., 2017)	90.4/135.1	21.6/33.12
(Sindagi and Patel, 2017)	**73.6/106.4**	20.1/30.1
Patch-Regression	83.6/131.5	**12.61**/23.6
Patch-Regression-Auxiliary Loss	81.9/127.9	13.2/**23.4**

Table 5.5: Comparison of various crowd counting techniques on the ShanghaiTech test set (Part A and B).

5.5 Joint Task Training

In this section the joint training of related crowd analysis tasks using a shared CNN is investigated. The tasks of crowd counting, crowd behaviour recognition and crowd density level estimation are trained and evaluated using the ShanghaiTech Part A, Violent-Flows and ShanghaiTech Density datasets respectively. As there is no common dataset annotated for all 3 analysis tasks this shared model must be trained in a round robin fashion, optimising one task in a given optimisation step before switching to the next. The loss function in use is also switched in and out in this round robin fashion. Single frame analysis is used for all tasks in order to simplify this set of experiments. The Resnet18 architecture is trained end-to-end with separate fully connected output layers for each task. This overall network configuration is illustrated in figure 5.3. The auxiliary output for density level estimation is included as it is shown to boost overall performance while the auxiliary output for crowd counting is omitted as it boosts performance on one dataset at the cost of another. Once a refined method has been developed using a collection of validation sets, comparisons are made to the leading techniques in the literature as well as the previously developed approaches presented in chapters 3 and 4. The following model selection issues are investigated for joint task training experiments:

- Task specific normalisation v. shared normalisation;
- Loss term weightings (equal v. manual v. data-driven);
- 2-task training v. 3-task training.

5.5.1 Task Specific Normalisation v. Shared Normalisation

The Resnet architecture performs batch normalisation (Ioffe and Szegedy, 2015) after each convolutional layer and is a major contributor towards the reliable convergence and predictive

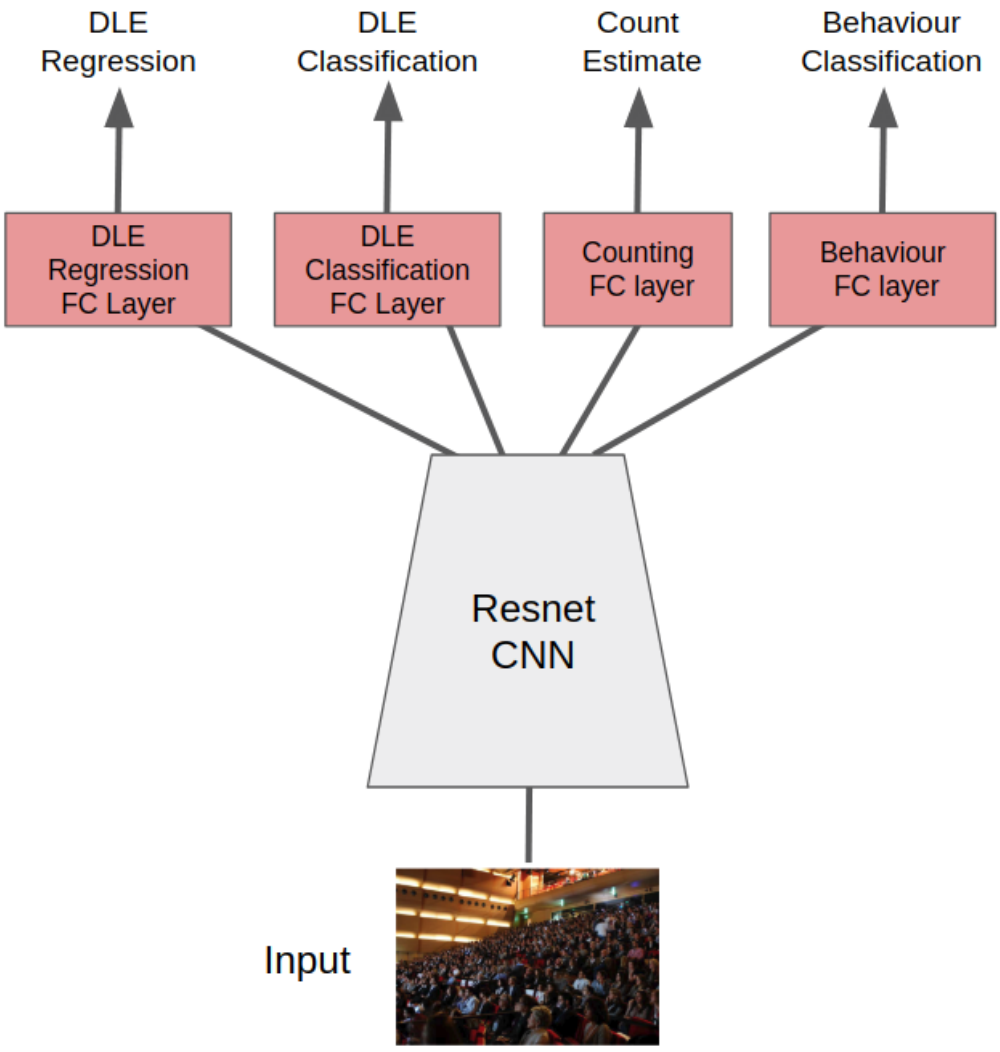

Figure 5.3: Multi-task crowd analysis architecture proposed for this set of experiments

performance achieved by this architecture. Several implementation strategies are investigated for normalisation in a multi-task configuration, including 1) Apply no task specific normalisation of any kind; 2) Include a task specific batch normalisation layer prior to the final fully connected layer(s) for each task; 3) Include a shared batch normalisation layer following the base network that is connected to all task specific layers. These 3 normalisation configurations are evaluated on a 2-task model trained jointly for behaviour recognition and crowd counting (3 task models are investigated in a later experiment). The results of this experiment are shown in table 5.6. Task specific batch normalisation results in the best overall performance, with a neg-

ligible increase in the overall model parameters from the inclusion of the additional BN layer. This configuration is used for all subsequent experiments.

Approach	Counting MAE/MSE	Behaviour ACC/AUC
No Additional BN	135.3/256.3	90.4%/0.943
Task specific BN	**132.50/241.3**	**91.6%/0.949**
Shared BN	166.8/305.4	89.5%/0.946

Table 5.6: Comparison of various batch normalisation strategies for multi-task crowd analysis. Evaluation is performed on the ShanghaiTech Part A validation set and the violent-Flows dataset (fold 1).

5.5.2 Loss Weightings

Various task weightings schemes are investigated for joint task training. Again a 2-task model trained for crowd counting and behaviour recognition is used for validation. While the use of round robin training results in optimisation not being performed in a truly joint fashion, loss weights can still be applied to the various tasks to influence training. The results of this experiment are shown in table 5.7. Equal loss weighting results in optimal performance for both tasks while the data driven weightings significantly degrade the performance in this case. Equal weighting across tasks is used for all subsequent joint training experiments.

5.5.3 2-Task v. 3-Task Training

Finally the impact of including additional crowd analysis tasks is investigated with crowd DLE added as this extra task. In order to ensure equal loss weighting across all tasks the weightings of the two DLE loss terms are each set to 0.5. End-to-end training is again carried out using a Resnet18 network initialised from a pre-trained ImageNet model. All other 2-task permutations

Scheme	Counting MAE/MSE	Behaviour ACC/AUC
Equal Weighting	**132.50/241.3**	**91.6%/0.949**
1.0/0.5	136.70/246.4	89.5%/0.946
0.5/1.0	139.50/255.7	90.5%/0.946
Data Driven	155.4/267.2	85.4%/0.927

Table 5.7: Comparison of various loss weighting for multi-task crowd analysis. Evaluation is performed on the ShanghaiTech Part A validation set and the violent-flows validation dataset

and single task baselines are also included for a comprehensive comparison. The results of this validation experiment are shown in figure 5.8. The inclusion of the 3rd task degrades performance for both crowd counting and behaviour recognition, resulting in poor performance across all tasks. All 2-task permutations result in performance boosts on the other hand. Therefore this may be a network capacity issue limiting the performance of the 3 task run.

Tasks	Counting MAE/MSE	Behaviour ACC/AUC	DLE MAE/ACC
Behaviour	N/A	89%/0.935	N/A
Counting	133.8/245.74	N/A	N/A
Density	N/A	N/A	**0.62/54%**
Behaviour/Counting	**132.50/241.3**	**91.6%/0.949**	N/A
Behaviour/Density	N/A	90.6%/0.939	0.62/54%
Counting/Density	134.8/249.8	N/A	0.62/54%
3-Task	237.2/467.7	68.7%/0.847	1.57/0.16

Table 5.8: Comparison of various multi-task training permutations for crowd analysis. Evaluation is performed on the ShanghaiTech part A validation set, the violent-flows validation set dataset and the ShanghaiTech density validation set.

5.5.4 Comparison With the State-Of-The-Art

Following the development of the proposed multi-task crowd analysis model comparisons are made with the leading techniques from the literature. As the proposed 3 task run on Resnet18 performs very poorly in validation, a 2 task run (behaviour recognition/crowd counting) is used. A 5-fold cross validation is used to compare performance on the violent-flows dataset, meaning 5 separate crowd counting runs are also generated, with the mean performance across these runs used for comparison purposes. Crowd density level estimation is not included in this experiment as there are no existing methods evaluated on the proposed ShanghaiTech density dataset in the literature. The results of this experiment are presented in table 5.9. Multi-task training boosts performance for both tasks, improving the mean accuracy on the violent-flows dataset beyond the proposed method and reducing the margin between the proposed counting model and the work of Sindagi and Patel (Sindagi and Patel, 2017) on the Shanghaitech Part A dataset. It is important to remember that the proposed counting technique comfortably outperforms Sindagi and Patel on ShanghaiTech Part B even in a single task training run.

Tasks	**Counting MAE/MSE**	**Behaviour ACC/AUC**
Behaviour-Single	N/A	95.2±7.5%/0.98
Counting-Single	83.62/131.5	N/A
(Senst et al., 2017)	N/A	93.12±8.7%/0.97
(Sindagi and Patel, 2017)	**73.6/106.4**	N/A
Behaviour/Counting	81.3/128.1	**95.4±6.9%/0.98**

Table 5.9: Comparison of the proposed multi-task crowd analysis model with the leading techniques in the literature for each task. Evaluation is performed on the ShanghaiTech Part A test set as well as the violent-Flows dataset (via a 5 fold cross validation). error margins are presented for the violent-flows dataset as is convention for this benchmarking task.

5.6 Discussion

Multi-task learning strategies have been shown to boost performance for several crowd analysis tasks with a negligible increase to model capacity required (a single additional fully connected layer per task). Auxiliary loss terms increase the benchmarking performance for density level estimation and for crowd counting in high congestion scenes. The joint training of related crowd analysis tasks has been shown to produce more accurate and robust predictions while reducing the number of required models parameters by 50% for a 2 task model. This holds true for 2 task joint training, while 3 task training results in inferior performance most likely due to a lack of model capacity within the fixed architecture used.

Task specific normalisation is shown to improve the predictive performance of multi-task crowd analysis models with a negligible increase in the overall parameter count. Various loss weighting schemes are investigated for multi-loss objective functions, with improved performance observed for crowd counting when loss weightings are informed by the gradient magnitudes calculated for the various network outputs. Equal weighting across tasks performs best for multi-objective density level estimation and the joint training of multiple analysis tasks. Varying the ratio of shared and task specific network parameters has been targeted as a possible direction for future work in area.

Performance boosts associated with multi-task learning are likely due to the degree of correlation between tasks, with complementary tasks benefiting the most. Uncorrelated tasks receive limited performance boosts when MTL is performed.

5.7 Summary

In this chapter deep learning based approaches to multi-task crowd analysis were developed. Various model selection issues including loss weightings, task specific normalisation and the number of tasks trained on simultaneously are investigated. State-of-the-art performance is achieved on the violent-flows dataset when joint training with crowd counting is performed. Issues that remain to be investigated for this task include the ratio of shared and task specific network parameters as well as the overall network capacity required to train 3 or more crowd analysis tasks simultaneously.

Chapter 6

Visual Domain Adaption In Object Counting

6.1 Introduction

This chapter investigates the use of domain adaptation (DA) techniques to extend deep learning models trained in a given visual domain to perform accurate analysis in other visual domains. Research question 3 is addressed by this chapter. The majority of the research to date for domain adaptation has focused on image classification tasks. This thesis however focuses on domain adaptation in regression tasks (i.e. object counting) which have up to now been largely unexplored. A new dataset for cell counting in microscopy is constructed and combined with existing collections for vehicle, person and wildlife counting. Recently proposed domain adaptation strategies are compared with more traditional techniques (feature extraction, fine tuning) both in terms of counting error and the number of new model parameters required to adapt a given network to a new domain. A domain classifier is also trained to distinguish between visual domains in the event that the observed domain is unknown during inference in a deployment

scenario. Once a refined multi-domain technique is developed, comparisons are made with the leading object counting methods in the literature for each domain. The work in this chapter was published at CVPR 2018 under the title "People, Penguins and Petri Dishes: Adapting Object Counting Models To New Visual Domains And Object Types Without Forgetting" (Marsden et al., 2018).

6.2 Contributions

The main contributions of this chapter are listed below:

- Rebuffi adapter modules are shown to be superior to traditional fine-tuning for domain adaptation in object counting;
- A cell counting dataset was constructed in collaboration the University College Dublin School Of Medicine;
- A multi-domain object counting is developed for crowd, cell, vehicle and wildlife counting.

6.3 Experimental Framework

As in the previous chapters, a common framework is used in which certain hyperparamaters and model selection choices are kept consistent across all experiments.

6.3.1 Fixed Hyperparameters

All of the fixed parameters for this set of experiments are kept consistent with those used in chapters 3, 4 and 5. The common framework used for all experiments in this chapter is listed in

table 6.1.

Optimiser	Adagrad (Duchi et al., 2011)
CNN Architecture	Resnet (18, 50 layers)
Regularisation	L_2 Weight Decay (0.001)
Augmentation	Random Crops, Random Flips
Initalisation	(Glorot and Bengio, 2010), bias terms set to 0
Objective Functions	MSE for object counting regression
Hardware	4GB Nvidia GTX 970 GPU, 8 core Intel i7 CPU, 32GB RAM

Table 6.1: Common framework used for all domain adaptation training runs.

6.3.2 Model Selection Issues Investigated

The following model selection issues are investigated for domain adaptation of object counting models:

- Traditional transfer learning methods v. recently proposed DA strategies
- Choosing the source domain used to initalise the model

6.4 Non-Crowd Object Counting Datasets

In order to carry out this research new counting datasets for other visual domains need to be utilised. Fortunately several fully labelled datasets from distinct visual domains have been made available to the research community. The TRANCOS dataset (Guerrero-Gómez-Olmedo et al., 2015) consists of 1244 images of vehicles with a mean object count of 36.5 and serves as the vehicle counting dataset for this study. The Penguins dataset (Arteta et al., 2016) contains 80,095

images of penguins with a mean object count of 7.18 and serves as the widlife counting dataset. Sample images from these datasets are shown in figure 6.1. Cell counting is another application where computer vision is a possible solution. The currently available cell counting datasets are either synthetic in nature (Xie et al., 2016) or captured via an imaging modality not commonly used for counting tasks in practice (e.g. histological slides (Cohen et al., 2017)). Cell counting in a tissue culture setting is a commonly used analysis tool for medical research and patient diagnoses, yet there is no publicly available dataset for developing computer vision solutions to this task. This need led to the development of a new cell counting dataset in collaboration with University College Dublin's School of Medicine.

Figure 6.1: Sample images taken from the TRANCOS (Guerrero-Gómez-Olmedo et al., 2015) and Penguins (Arteta et al., 2016) datasets.

6.5 Cell Counting Dataset Construction

There is no fully annotated dataset suitable for the development of cell counting methods in a tissue culture setting. To address this the Dublin Cell Counting (DCC) dataset is developed. This dataset consists of 177 images containing a wide array of tissue and species. Amongst them

are examples of stem cells derived from embryonic mice, isolated human lung adenocarcinoma and examples of primary human monocytes isolated from a healthy human volunteer. Several factors were varied during image capture to provide a more representative set of images. First, the density of cells loaded onto the slide naturally varies as cell lines proliferate at different rates. Second, the morphology and size of the cells for each cell line can vary significantly. Furthermore, the objective lens used during imaging was varied as was the diameter of the diaphragm which controls the amount of light hitting the sample. Finally, the haemocytometer grid size was varied to produce a representative set of non-cellular image artifacts. Cell images were obtained via a camera mounted on an Olympus CKX41 microscope using both $4\times$ and $10\times$ objectives. The high levels of variation in this collection provides a representative and challenging benchmark.

Once the full set of image is acquired an annotation process was performed by a domain expert with a background in molecular biology, applying a single pixel dot to each cell within a given image. The mean cell count across these images is 34.1 with a standard deviation of 21.8, showing the significant variation in cell density level. 100 images are used for training and validation while the remaining 77 form an unseen test set. Sample images from this newly created dataset are shown in figure 6.2. Combining this collection with the TRANCOS, Penguins and ShanghaiTech datasets provides 4 high quality object counting datasets in distinct visual domains which can be used for domain adaptation research.

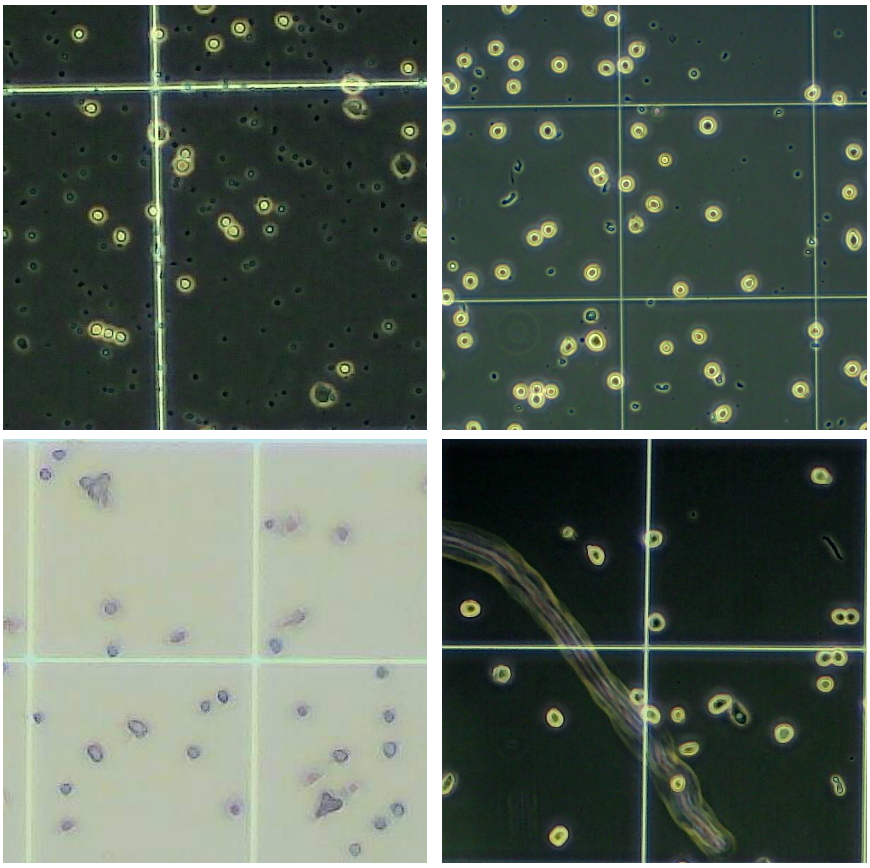

Figure 6.2: DCC dataset examples showing the significant variation within this collection.

6.6 Domain Adaptation Methods

6.6.1 Traditional Transfer Learning v. New DA Strategies

In this section more traditional transfer learning techniques (fine-tuning, feature extraction) are compared with a recently proposed domain adaptation strategy for the task of object counting. The recently proposed DA strategy evaluated is the residual adapter modules of Rebuffi et al. (Rebuffi et al., 2017). This approach includes a set of so called *adapter modules* before each convolutional or fully connected layer which allow the network to adapt to the statistical properties of each visual domain. These domain-specific modules are interchanged during training and inference depending on the chosen visual domain (this switching concept is highlighted in figure 6.3). Each adapter module (visualised in figure 6.4) consists of a batch normalisation

layer, a bank of N 1×1 convolutional filters (where N is the length of the input vector), a skip connection to enable residual representation learning followed by a final batch normalisation layer. 1×1 convolutions are applicable in the case of vector data. These adapter modules contribute only a small percentage to the overall parameter count of a given CNN.

After an object counting model has been trained end-to-end for a given domain (including a set of domain adapter modules for that domain) it can be adapted to another domain by training a fresh set of adapter modules and a new fully connected output layer, with the shared model parameters frozen during optimisation. This allows for sequential model training to be performed with performance in the original domain preserved. The original domain is referred to as the *source* domain while the newly added domain is referred to as the *target* domain. The benefits of this approach are the potential for transfer learning and a significant reduction in the overall parameter count compared to an ensemble of single domain models.

In this chapter this adapter module approach is compared to several combinations of fine-tuning and feature extraction as well as training in the target domain from scratch. A Resnet18 network is used to perform patch-based regression counting for all runs, with training carried out for 10,000 iterations each time. Comparisons are made both in terms of MAE performance and the number of new model parameters that need to be introduced for the target domain. For this experiment, cell counting is used as the source domain while crowd counting is used as the target domain (the choice of source and target is fully explored in later experiments). Performance is evaluated on the ShanghaiTech dataset part A validation set. The results of this experiment are shown in table 6.2. The use of the adapter modules matches the performance of training in the target domain from scratch and does so while introducing just 500k additional parameters (compared to the 11.1M required when training from scratch in the target domain). Adapter modules significantly outperform all fine-tuning/feature-extraction runs in terms of the

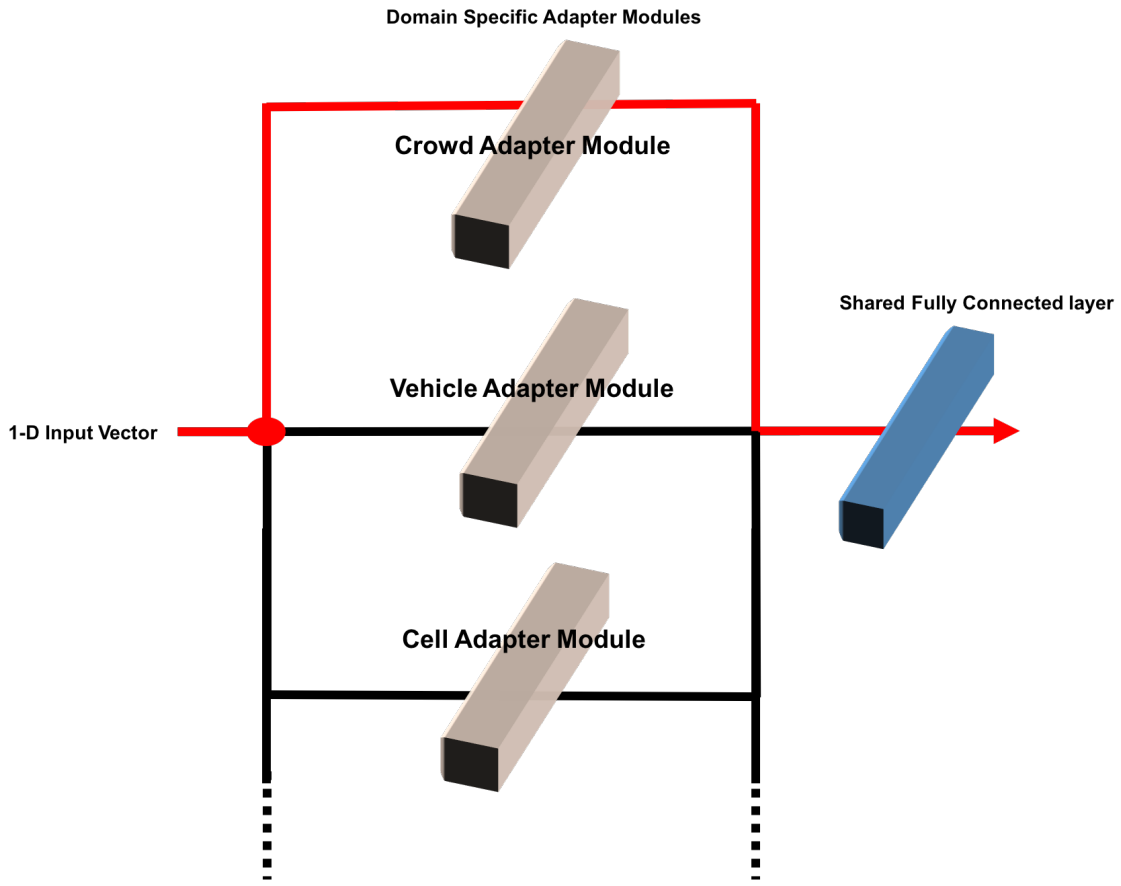

Figure 6.3: Domain specific adapter modules of Rebuffi *et al.* (Rebuffi et al., 2017) are interchanged during training and inference depending on the chosen counting domain (red path).

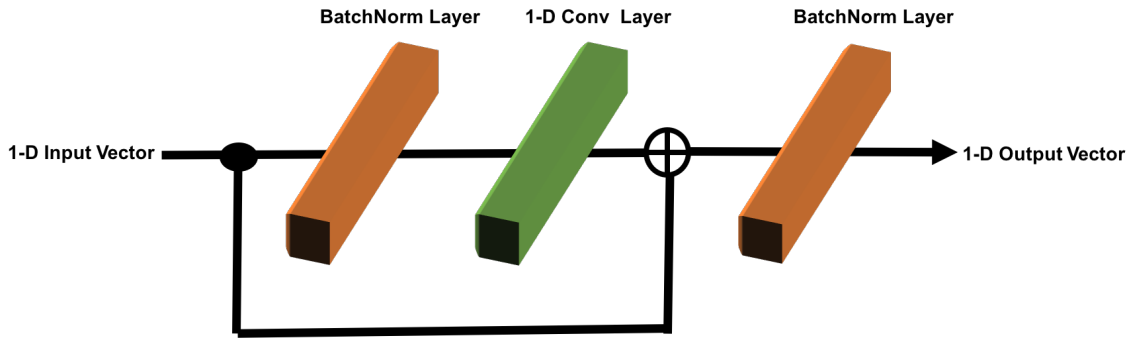

Figure 6.4: The residual adapter module (Rebuffi et al., 2017).

performance gained relative to new parameters required. This technique will therefore be used for all subsequent object counting domain adaptation experiments. Adapter modules and other refined approaches to DA can be viewed as an evolution of traditional transfer learning methods.

Approach	Target Domain Parameters	Target MAE
Target From Scratch	11.1M	133.8
Final 9 layers fine-tuned	10.2M	145.3
Final 5 layers fine-tuned	8.3M	232.2
Final 2 layers fine-tuned	4.7M	245.5
Adapter Modules (Rebuffi et al., 2017)	0.5M	**133.5**

Table 6.2: MAE validation performance on the Shanghaitech dataset (part A) for various domain adaptation strategies. Cell counting (via the DCC dataset) is used as the source domain for each run. For all fine-tuning runs the non-trained layers are frozen after training on the source domain.

6.6.2 Choosing the Source Domain

This section investigates the choice of source domain when performing domain adaptation for object counting. Performance is evaluated across all 4 visual domains introduced earlier (cells, crowds, vehicles, penguins). After training on a given source domain the model is then adapted to the other 3 as targets. Rebuffi's DA method (Rebuffi et al., 2017) is utilised for all runs. The Resnet18 architecture is used for all runs, with training carried out for 10,000 iterations each time. The validation set for each domain dataset is used to measure performance. Table 6.3 presents the MAE score observed for each permutation, with each row corresponding to the source domain and each column corresponding to the target domain. The diagonal entries correspond to the performance achieved when training the network from scratch for each domain. Another run is also included where concurrent training is performed for all domains in a round robin fashion.

It can be seen that using the cell domain as the source results in the best overall performance, achieving superior MAE on 3 of the 4 domains and beating the concurrent training run. Concur-

Visual Domain	Crowd (Target)	Vehicles (Target)	Wildlife (Target)	Cells (Target)
Crowd (Source)	133.5	10.3	6.8	12.9
Vehicles (Source)	149.2	**9.9**	6.1	11.5
Wildlife (Source)	146.3	10.3	6.05	10.2
Cells (Source)	**131.2**	10.2	**5.7**	**9.5**
Concurrent Training	135.6	9.95	5.93	10.1

Table 6.3: The MAE validation performance achieved when varying the source domain. A concurrent training run is also included.

rent training likely suffers from trying to balance all 4 domains to find an optimal function for all domains. Sequential training via Rebuffis method on the other hand allows for each domain to be optimised individually. Adapting from the cell domain achieves performance superior to training from scratch on both the crowd and wildlife domains, which is noteworthy given the small number of domain-specific parameters trained. The high performance observed across domains when adapting from a cell counting model is likely due to the significant morphological variation (observed qualitatively) between cell objects in the DCC dataset, resulting in a more varied set of learned features which can be applied to a variety of counting tasks. Future work will look to measure this morphological variation. In all subsequent experiments the cell domain is used as the source domain.

6.7 Domain Classification

If the visual domain observed during inference is unknown this can be predicted by extending the multi-domain counting network to also perform domain classification. To accomplish this the final fully connected layer is interchanged with a K-neuron fully connected layer, where K

is the number of visual domains considered. Following this final layer, a softmax activation is applied. Training can then be performed by again freezing the shared model parameters and creating a fresh set of adapter modules for the domain classification task. Categorical cross entropy is minimised to train this classifier. The overall domain classification concept is visualised in figure 6.5.

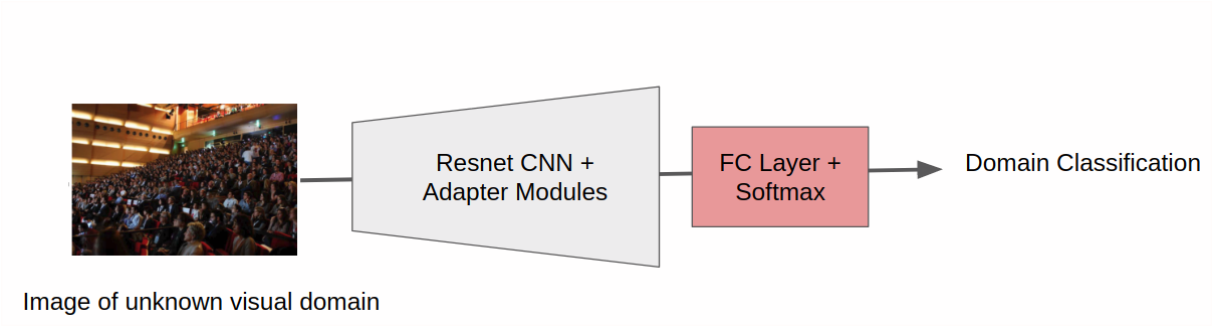

Figure 6.5: Domain classification pipeline adapted from an existing multi-domain object counting model.

A representative dataset is constructed to train this domain classifier by taking 300 images from the dataset of each of the 4 visual domains used in this study. Horizontal flip augmentation is applied to provide 300 cell images for the smaller DCC dataset. This 1200 image dataset is then divided into a training, validation and test set using a 7:1:2 split while ensuring equal representation among visual domains. The cell domain is again used as the source domain and training is carried out for 10,000 iterations. Table 6.4 compares validation accuracy for several configurations including: training from scratch, adapting from a cell counting network using a fresh set of adapter modules and adapting from a cell counting network and just training the final fully connected layer. The best overall validation performance is observed when training the full network from scratch for domain classification, however this performance is closely followed by the run which adds only a set of adapter modules (+0.5M parameters) to an existing model. Both of these approaches significantly outperform the run which trains only the final fully connected layer, again highlighting the potential of using a distributed domain adaptation technique such as

Rebuffi's method. Test set using of 99.2% is achieved using the adapter module configuration, showing how distinct and easily distinguished these visual domains are. With this approach it is possible to combine a multi-domain object counting model with a domain classifier into a single network while achieving strong performance across all tasks.

Source Domain	Training Approach	Accuracy
None	Entire Network Trained	**99.6%**
Cells	Adapter Modules + Final Layer	98.1%
Cells	Final Layer Only	58.3%

Table 6.4: Domain classification validation accuracy as the training approach is varied.

6.8 Comparison to the State-of-the-art

The developed multi-domain object counting technique is now compared to the leading techniques from the literature for each visual domain. Evaluation is performed in all cases on the relevant test set. The cell domain is used as the source in all cases. Table 6.5 compares crowd counting performance on the ShanghaiTech dataset. The multi-domain model achieves performance comparable to the state-of-the-art run proposed in chapter 4. Table 6.6 then compares vehicle counting performance on the TRANCOS test set, with competitive performance achieved and boosted by the use of a multi-domain object counting strategy.

Wildlife counting test performance on the Penguins dataset is presented in table 6.7, with state-of-the-art performance achieved on this benchmark and multi-domain counting again boosting performance. MAE is computed on the Penguin test set with respect to the max count label for each image (as there are multiple annotators). The separate site dataset split is used for the Penguins collection and no depth information is utilised. Finally table 6.8 compares cell

	Part A		Part B	
Method	MAE	MSE	MAE	MSE
(Zhang et al., 2016)	110.2	173.2	26.4	41.3
(Sam et al., 2017)	90.4	135.0	21.6	33.12
(Sindagi and Patel, 2017)	**73.6**	**106.4**	20.1	30.1
Single-Domain Model (Proposed)	83.62	131.5	**12.61**	**23.6**
Multi-Domain Model (Proposed)	84.1	133.6	13.12	24.3

Table 6.5: Comparing the performance of various crowd counting approaches on the Shanghaitech dataset including the developed multi-domain counting model.

Method	MAE
(Onoro-Rubio and López-Sastre, 2016)	10.99
(Zhang et al., 2017)	**4.2**
Single-Domain Model (Proposed)	9.7
Multi-Domain Model (Proposed)	9.5

Table 6.6: Comparing performance of various vehicle counting approaches on the TRANCOS test set.

counting performance on the MBN dataset (which uses images from histological slides rather than from a tissue culture setting). Competitive performance is achieved by the proposed multi-domain method

.

Overall the proposed multi-domain counting network achieves state-of-the-art performance in crowd and wildlife counting with competitive performance in cell and vehicle counting. This strong overall performance is achieved using a common framework with a dramatic reduction in the overall parameter count. This method can be extended to perform other counting tasks

Method	MAE
(Arteta et al., 2016)	8.11
Single-Domain Model (Proposed)	6.1
Multi-Domain Model (Proposed)	**5.8**

Table 6.7: Comparing performance of various counting techniques on the Penguins dataset test set. MAE is computed with respect to the max count on each image (as there are multiple annotators). The separate site dataset split is used and no depth information is utilised.

Method	N=5	N=10	N=15
(Xie et al., 2016)	28.9 ± 22.6	22.2± 11.6	21.3± 9.4
Multi-Domain Model (Proposed)	23.6± 4.6	21.5± 4.2	20.5± 3.5
(Cohen et al., 2017)	**12.6± 3.0**	**10.7± 2.5**	**8.8±2.3**

Table 6.8: Cell counting MAE performance on the MBM dataset. Out of the 44 images in this collection, N are used for training, N for validation and an unseen 14 images for testing. At least 10 runs using random dataset splits are performed for the each N value.

over time once a labelled dataset is provided.

6.9 Discussion

The benefits of a multi-domain object counting method are demonstrated in this chapter. Benchmarking performance consistent with and sometimes exceeding single-domain baselines can be achieved while significantly reducing the overall parameter count. The superiority of newer domain adaptation methods such as the work of Rebuffi et al. (Rebuffi et al., 2017) over traditional transfer learning is also shown. State-of-the-art performance is achieved in crowd and wildlife counting while competitive performance is observed in vehicle and cell counting. This dispar-

ity in performance is most likely due to the fixed framework used for all domains, with model selection issues such as the patch size used during training and inference kept at a constant for all domains.

Future work in this area will investigate alternate domain adaptation strategies, the domain-specific optimisation of hyperparamters such as patch size, choices in network architecture and the inclusion of auxiliary loss terms. Domain adaptation can also be investigated for other crowd analysis tasks such as behaviour recognition in video and crowd density level estimation.

6.10 Summary

In this chapter a deep learning approach to multi-domain object counting was developed. A recently proposed domain adaptation technique was applied to the task of object counting and shown to be superior to more traditional transfer learning methods such as feature extraction and fine tuning. Strong performance across all domains was observed with best in class performance in people and wildlife counting achieved using the shared model. A new dataset for cell counting was constructed in collaboration with researchers from University College Dublin. This multi-domain approach also results in a significant reduction in overall model parameters and can be extended over time to perform object counting in new visual domains.

Chapter 7

Conclusions

This chapter summarises the research carried out as part of this thesis. Each of the hypotheses proposed in chapter 1 are discussed with respect to the experimental results produced in chapters 3-6. The outcomes of this experimental work are then presented as a set of core research contributions. Finally some potential directions for future work are proposed before some closing remarks.

7.1 Hypotheses

This section discusses each of the hypotheses proposed as part of this thesis with respect to the experimental results produced in chapters 3-6.

Hypothesis 1

Data-driven models such as convolutional neural networks are superior to hand-crafted methods for vision-based crowd analysis tasks both in terms of predictive performance and adaptability to various problem types.

Experiments conducted in chapters 3 and 4 investigate the application of deep learning meth-

ods, specifically convolutional neural networks, to the area of vision-based crowd analysis. A refined CNN technique is developed for four crowd analysis tasks through extensive validation before comparisons are made to techniques which rely on hand crafted features as well as the leading methods from the literature, with the DL approaches consistently outperforming other methods.

Chapter 3 investigates crowd behaviour analysis, which itself is separated into two tasks, crowd behaviour recognition and crowd behaviour anomaly detection. For the task of crowd behaviour recognition the superiority of the refined CNN approach over hand crafted methods is demonstrated on multiple datasets, with a 2% improvement in mean accuracy on the Violent-Flows dataset and a 28% relative improvement in mean AUC score on the more challenging WWW Crowd dataset. When compared to the leading techniques, the proposed CNN method achieves state-of-the-art crowd behaviour recognition performance on the Violent-flows dataset and competitive performance on the WWW crowd (1% inferior mean AUC score). For the task of crowd behaviour anomaly detection the proposed CNN method improves upon the leading hand-crafted approach with a 31% relative increase in AUC score on the LV dataset. This method is also superior to all other approaches from the literature evaluated on the LV dataset. These significant improvements show that data-driven approaches clearly lead to superior performance in the very challenging field of crowd behaviour analysis. Through the extensive experimentation carried out in chapter 3 the best practices for implementing a vision-based crowd behaviour analysis system can be summaried as follows:

- The use of a data-driven model such as a CNN;
- The analysis of longer term temporal dynamics observed over a 100+ frame period;
- The joint analysis of spatial and temporal dynamics in video

- The joint analysis of RGB and optical flow channels to capture local appearance and motion features.

Chapter 4 investigates crowd congestion analysis, which itself is divided into crowd counting and crowd density level estimation. For the task of crowd counting the performance gains associated with the refined CNN method over hand crafted features are demonstrated on the ShanghaiTech dataset, with a 51% reduction in MAE observed on Part A of the dataset and an 82% reduction in MAE on Part B. This proposed CNN method achieves state-of-the-art crowd counting performance on the ShanghaiTech dataset. For crowd density level estimation an improvement in accuracy of 15% and a 32% reduction in MAE is observed on the newly proposed ShanghaiTech Density dataset when comparing the proposed deep learning approach to a hand-crafted baseline. These performance gains can be increased to 40% and 70% respectively by re-purposing a trained crowd counting model for density level estimation, albeit with a significant increase in the computational requirements. Again these significant increases in benchmarking performance over hand-crafted approaches highlight the superiority of data-driven approaches to vision-based crowd congestion analysis. Through extensive experimentation in chapter 4 the best practices for implementing a crowd congestion analysis system can the summarised as follows:

- The use of a data-driven model such as a CNN;
- A patch-based regression approach to crowd counting;
- Random image cropping during the training of a crowd counting model in order to increase data variation and boost model generalisation;
- The use of a classification approach to crowd DLE rather than a regression + estimation rounding method.

Considering the overall set of results from chapters 3 and 4 it has been shown that the use of data-driven approaches leads to vastly superior performance in vision-based crowd analysis over hand crafted approaches.

Hypothesis 2

Multi-task learning techniques can be used to improve the predictive performance of vision-based crowd analysis models and reduce the overall trainable parameter count across related crowd analysis tasks.

Experiments carried out in chapter 5 investigate the use of MTL techniques for vision-based crowd analysis. Auxiliary loss terms are firstly investigated as a means to improve single task performance before the joint training of several related crowd analysis tasks is evaluated.

Auxiliary loss terms are shown to improve single-task performance for multiple crowd analysis problems. For crowd density level estimation a 2% improvement in accuracy and a 1% reduction in MAE is observed when including an auxiliary regression loss term to a classification based model, with a negligible increase in the number of network parameters and inference time. For crowd counting, the inclusion of an auxiliary heatmap generation loss to a patch-based regression model results in a 3% reduction in MAE on ShanghaiTech part A and a 4% increase in MAE on part B. Part A contains high congestion scenes and benefits more from the inclusion of the heatmap loss, while Part B contains lower density scenes and suffers when it is used. Therefore the inclusion of such an auxiliary counting loss boosts performance in high congestion scenes at the cost of performance in lower congestion scenes. This may be suitable in certain scenarios and applications. Overall the inclusion of auxiliary loss terms has been shown to boost the performance of crowd congestion analysis systems with a negligible increase in computational cost.

The joint training of multiple related crowd analysis tasks is also shown to boost the overall predictive performance of crowd analysis tasks though with several limitations. Jointly training a 2-task model for crowd counting and behaviour recognition reduces the overall parameter count by 50% and results in a 2% reduction in MAE for crowd counting and a 0.2% increase in mean accuracy for crowd behaviour recognition. Including a third task however leads to significantly inferior performance most likely due to the fixed capacity of the network used, resulting in the model underfitting. Including additional model capacity could help address this issue but goes against the parameter reduction goal of this research task. Equal task loss weightings during optimisation and task-specific batch normalisation prior to each output layer are observed to be some overall best practices when training a multi-task crowd analysis models.

Overall multi-task learning strategies have been shown to increase the predictive performance of crowd analysis tasks while reducing the overall parameter count during joint task training. This strategy can however fail when the capacity of a given model cannot cover the N tasks being trained.

Hypothesis 3

Domain adaptation techniques can be used to extend a crowd analysis model to other visual domains and vice versa while retaining model accuracy for all domains and significantly reducing the overall parameter count.

Experiments carried out in chapter 6 address the application of domain adaption techniques to extend crowd analysis models to other domains and vice versa. This concept of domain adaptation is investigated for visual object counting for the first time. A new dataset for cell counting in a tissue culture setting, referred to as the Dublin Cell Counting dataset, is constructed to help conduct this research. This newly proposed dataset is combined with existing collections for

crowd, vehicle and wildlife counting to develop a multi-domain counting model through adaptation. The recently developed domain adaptatation strategy of Rebuffi *et al.* (Rebuffi et al., 2017) is compared to more traditional transfer learning strategies (fine-tuning, feature extraction).

The Rebuffi method is first used to adapt a cell counting model trained on DCC (source) to perform crowd counting on ShanghaiTech Part A (target). The use of this recently proposed DA method results in superior MAE performance on ShanghaiTech Part A compared to all traditional transfer learning runs (fine-tuning, feature extraction) and closely matches the performance when training from scratch on the target domain. Another benefit of this approach is the small increase in the overall parameter count (550,000) for each additional domain compared to the 11.1M additional parameters required when training each new domain from scratch. The choice of which source domain to use when developing a multi-domain object counting model is then investigated, with the best overall performance achieved when the cell domain is used as the source. This is likely to be due to the significant morphological variation observed in the DCC dataset, leading to a broader and more general set of learned features. In terms of predictive performance, the developed multi-domain model achieves state-of-the-art MAE on the ShanghaiTech dataset (crowd) and Penguins Dataset (wildlife) while competitive performance is observed on the MBN (cell) and TRANCOS (vehicle) datasets. The downsides of this approach are the static framework used for all counting tasks in all domains, with model selection issues such as the patch size used during training and inference fixed across all domains.

Overall it has been demonstrated that DA techniques can be used to extend crowd analysis models to new visual domains and vice versa, with no drop in predictive performance and a significant reduction in model parameters when developing a multi-domain analysis model. Despite the shortcomings of the current implementation, there is great potential for domain adaptation in fields such as object counting.

7.2 Research Contributions

The core research contributions of this thesis can be summarised as follows:

- A 3D Late Fusion CNN approach is developed for crowd behaviour recognition;
- The trained model is used to perform distance based crowd behaviour anomaly detection on the LV Dataset;
- The proposed technique is shown to be superior to a hand crafted baseline for both behaviour recognition and anomaly detection;
- State-of-the-art performance is achieved on the LV and Violent-Flows datasets;
- A patch-based regression approach to crowd counting is developed;
- A crowd density level estimation dataset is constructed;
- The proposed technique is shown to be superior to a hand crafted baseline for both crowd counting and crowd density level estimation;
- State-of-the-art performance is achieved on the ShanghaiTech crowd counting dataset;
- Auxiliary loss functions are shown to improve crowd density level estimation performance with a negligible increase to the overall parameter count;
- A joint model for crowd counting and behaviour recognition is developed, improving the predictive performance of both tasks and reducing the overall parameter count by 50%;
- Rebuffi adapter modules are shown to be superior to traditional fine-tuning for domain adaptation in object counting;

- A cell counting dataset was constructed in collaboration with the University College Dublin School Of Medicine;
- A multi-domain object counting is developed for crowd, cell, vehicle and wildlife counting.

7.3 Future Work

Future research in the area of vision based crowd analysis can proceed along any of the following directions:

- The utilisation of depth information to help boost the performance of crowd analysis models;
- The combination of multi-frame and single-frame analysis pipelines using MTL techniques to perform tasks such as multi-frame crowd behaviour recognition and single-frame crowd counting in a shared model;
- The construction of a multi-frame crowd counting dataset to investigate video-based crowd counting;
- The localisation of crowd behaviour anomalies within large and complex scenes using deep learning methods;
- Domain-specific optimisation of model configuration choices such as input patch size and model architecture when developing multi-domain object counting models;
- A combination of regression-based object counting and bounding box object detection that can adjust based on the content of the scene or scene region;

- An investigation into domain adaptation for video recognition tasks using multi-frame models.
- The visualisation of trained CNN features used in domain adaptation experiments.
- The use of generative models for crowd dataset augmentation.
- The measurement of morphological variation in object counting datasets.

7.4 Closing Remarks

Vision-based crowd analysis is just one of the many research topics to benefit from the application of data-driven models, deep neural networks and hardware accelerated numerical optimisation. In this thesis the superiority of these methods over more traditional hand-crafted methods is demonstrated and quantified for crowd analysis problems. While significant progress has been made, a great deal of work must be done before benchmarking performance is high enough for real world deployment of these systems to be considered. Another challenge preventing real world deployment of these crowd analysis models is the high computational demands of CNN models. Multi-task learning and domain adaptation methods can reduce the overall parameter count of these models and in turn the memory requirements when employing CNN models for crowd analysis. Deployable crowd analysis via deep learning techniques will likely come through a combination of hardware advancements, model compression methods, dataset generation and transfer learning techniques.

Bibliography

Arteta, C., Lempitsky, V. and Zisserman, A. (2016), Counting in the wild, *in* 'European Conference on Computer Vision', Springer, pp. 483–498.

Biswas, S. and Babu, R. V. (2013), Real time anomaly detection in h. 264 compressed videos, *in* 'Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on', IEEE, pp. 1–4.

Boiman, O. and Irani, M. (2007), 'Detecting irregularities in images and in video', *International journal of computer vision* **74**(1), 17–31.

Carreira, J. and Zisserman, A. (2017), Quo vadis, action recognition? a new model and the kinetics dataset, *in* '2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', IEEE, pp. 4724–4733.

Caruana, R. (1998), Multitask learning, *in* 'Learning to learn', Springer, pp. 95–133.

Chen, K., Loy, C. C., Gong, S. and Xiang, T. (2012), Feature mining for localised crowd counting., *in* 'BMVC', Vol. 1, p. 3.

Cohen, J. P., Lo, H. Z. and Bengio, Y. (2017), 'Count-ception: Counting by fully convolutional redundant counting', *arXiv preprint arXiv:1703.08710* .

Csurka, G. (2017), 'Domain adaptation for visual applications: A comprehensive survey', *arXiv preprint arXiv:1702.05374* .

Dalal, N. and Triggs, B. (2005), Histograms of oriented gradients for human detection, *in* 'Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on', Vol. 1, IEEE, pp. 886–893.

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Senior, A., Tucker, P., Yang, K., Le, Q. V. et al. (2012), Large scale distributed deep networks, *in* 'Advances in neural information processing systems', pp. 1223–1231.

Dee, H. M. and Caplier, A. (2010), Crowd behaviour analysis using histograms of motion direction, *in* 'Image Processing (ICIP), 2010 17th IEEE International Conference on', IEEE, pp. 1545–1548.

Duan, L., Tsang, I. W., Xu, D. and Maybank, S. J. (2009), Domain transfer svm for video concept detection, *in* 'Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on', IEEE, pp. 1375–1381.

Duchi, J., Hazan, E. and Singer, Y. (2011), 'Adaptive subgradient methods for online learning and stochastic optimization', *Journal of Machine Learning Research* **12**(Jul), 2121–2159.

Evgeniou, T. and Pontil, M. (2004), Regularized multi–task learning, *in* 'Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining', ACM, pp. 109–117.

Farnebäck, G. (2003), Two-frame motion estimation based on polynomial expansion, *in* 'Scandinavian conference on Image analysis', Springer, pp. 363–370.

Fu, M., Xu, P., Li, X., Liu, Q., Ye, M. and Zhu, C. (2015), 'Fast crowd density estimation with convolutional neural networks', *Engineering Applications of Artificial Intelligence* **43**, 81–88.

Funahashi, K.-i. and Nakamura, Y. (1993), 'Approximation of dynamical systems by continuous time recurrent neural networks', *Neural networks* **6**(6), 801–806.

Ge, W. and Collins, R. T. (2009), Marked point processes for crowd counting, *in* 'Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on', IEEE, pp. 2913–2920.

Gibson, J. J. (1950), 'The perception of the visual world.'.

Glorot, X. and Bengio, Y. (2010), Understanding the difficulty of training deep feedforward neural networks., *in* 'Aistats', Vol. 9, pp. 249–256.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A. and Bengio, Y. (2013), 'An empirical investigation of catastrophic forgetting in gradient-based neural networks', *arXiv preprint arXiv:1312.6211* .

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014), Generative adversarial nets, *in* 'Advances in neural information processing systems', pp. 2672–2680.

Guerrero-Gómez-Olmedo, R., Torre-Jiménez, B., López-Sastre, R., Maldonado-Bascón, S. and Oñoro-Rubio, D. (2015), Extremely overlapping vehicle counting, *in* 'Iberian Conference on Pattern Recognition and Image Analysis', Springer, pp. 423–431.

Han, K., Wan, W., Yao, H. and Hou, L. (2017), 'Image crowd counting using convolutional neural network and markov random field', *arXiv preprint arXiv:1706.03686* .

Hassner, T., Itcher, Y. and Kliper-Gross, O. (2012*a*), Violent flows: Real-time detection of violent crowd behavior, *in* 'Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on', IEEE, pp. 1–6.

Hassner, T., Itcher, Y. and Kliper-Gross, O. (2012*b*), Violent flows: Real-time detection of violent crowd behavior, *in* '2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops', Ieee, pp. 1–6.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6239348*

He, K., Zhang, X., Ren, S. and Sun, J. (2016), Deep residual learning for image recognition, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 770–778.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. (2017), 'Mobilenets: Efficient convolutional neural networks for mobile vision applications', *arXiv preprint arXiv:1704.04861* .

Hsu, C.-L. and Lin, J. C.-C. (2016), 'An empirical examination of consumer adoption of internet of things services: Network externalities and concern for information privacy perspectives', *Computers in Human Behavior* **62**, 516 – 527.
URL: *http://www.sciencedirect.com/science/article/pii/S0747563216302990*

Hu, Y., Chang, H., Nian, F., Wang, Y. and Li, T. (2016), 'Dense crowd counting from still images with convolutional neural networks', *Journal of Visual Communication and Image Representation* **38**, 530–539.

Idrees, H., Saleemi, I., Seibert, C. and Shah, M. (2013), Multi-source multi-scale counting in extremely dense crowd images, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 2547–2554.

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A. and Brox, T. (2017), 'Flownet 2.0: Evolution of optical flow estimation with deep networks', *CVPR* .

Ioffe, S. and Szegedy, C. (2015), 'Batch normalization: Accelerating deep network training by reducing internal covariate shift'.

Ji, S., Xu, W., Yang, M. and Yu, K. (2013), '3d convolutional neural networks for human action recognition', *IEEE transactions on pattern analysis and machine intelligence* **35**(1), 221–231.

Jiang, W., Zavesky, E., Chang, S.-F. and Loui, A. (2008), Cross-domain learning methods for high-level visual concept classification, *in* 'Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on', IEEE, pp. 161–164.

Kendall, A., Gal, Y. and Cipolla, R. (2017), 'Multi-task learning using uncertainty to weigh losses for scene geometry and semantics', *arXiv preprint arXiv:1705.07115* .

Kocev, D., Vens, C., Struyf, J. and Džeroski, S. (2007), Ensembles of multi-objective decision trees, *in* 'European Conference on Machine Learning', Springer, pp. 624–631.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* 'Advances in neural information processing systems', pp. 1097–1105.

Krogh, A. and Hertz, J. A. (1992), A simple weight decay can improve generalization, *in* 'Advances in neural information processing systems', pp. 950–957.

LeCun, Y., Touresky, D., Hinton, G. and Sejnowski, T. (1988), A theoretical framework for back-propagation, *in* 'Proceedings of the 1988 connectionist models summer school', CMU, Pittsburgh, Pa: Morgan Kaufmann, pp. 21–28.

Leyva, R., Sanchez, V. and Li, C.-T. (2017), The lv dataset: A realistic surveillance video dataset for abnormal event detection, *in* 'Biometrics and Forensics (IWBF), 2017 5th International Workshop on', IEEE, pp. 1–6.

Li, M., Zhang, Z., Huang, K. and Tan, T. (2008), Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection, *in* 'Pattern Recognition, 2008. ICPR 2008. 19th International Conference on', IEEE, pp. 1–4.

Li, Y., Zhang, X. and Chen, D. (2018), 'Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes', *arXiv preprint arXiv:1802.10062* .

Li, Z. and Hoiem, D. (2017), 'Learning without forgetting', *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Lloyd, K., Rosin, P. L., Marshall, D. and Moore, S. C. (2017), 'Detecting violent and abnormal crowd activity using temporal analysis of grey level co-occurrence matrix (glcm)-based texture measures', *Machine Vision and Applications* **28**(3-4), 361–371.

Lowe, D. G. (2004), 'Distinctive image features from scale-invariant keypoints', *International journal of computer vision* **60**(2), 91–110.

Lu, C., Shi, J. and Jia, J. (2013), Abnormal event detection at 150 fps in matlab, *in* 'Computer Vision (ICCV), 2013 IEEE International Conference on', IEEE, pp. 2720–2727.

Lu, Y., Kumar, A., Zhai, S., Cheng, Y., Javidi, T. and Feris, R. (2017), Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 5334–5343.

Ma, W., Huang, L. and Liu, C. (2008), Crowd estimation using multi-scale local texture analysis and confidence-based soft classification, *in* 'Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on', Vol. 1, IEEE, pp. 142–146.

Mahadevan, V., LI, W.-X., Bhalodia, V. and Vasconcelos, N. (2010), Anomaly detection in crowded scenes, *in* 'Proceedings of IEEE Conference on Computer Vision and Pattern Recognition', pp. 1975–1981.

Mallya, A. and Lazebnik, S. (2017), 'Packnet: Adding multiple tasks to a single network by iterative pruning', *arXiv preprint arXiv:1711.05769* .

Mallya, A. and Lazebnik, S. (2018), 'Piggyback: Adding multiple tasks to a single, fixed network by learning to mask', *arXiv preprint arXiv:1801.06519* .

Marana, A. N., Costa, L. D. F., Lotufo, R. and Velastin, S. A. (1999), Estimating crowd density with minkowski fractal dimension, *in* 'Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on', Vol. 6, IEEE, pp. 3521–3524.

Marsden, M., McGuinness, K., Little, S., Keogh, C. E. and O'Connor, N. E. (2018), 'People, penguins and petri dishes: adapting object counting models to new visual domains and object types without forgetting'.

Marsden, M., McGuinness, K., Little, S. and O'Connor, N. E. (2016), 'Fully convolutional crowd counting on highly congested scenes', *arXiv preprint arXiv:1612.00220* .

Marsden, M., McGuinness, K., Little, S. and O'Connor, N. E. (2017), Resnetcrowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification, *in* 'Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on', IEEE, pp. 1–7.

Masters, D. and Luschi, C. (2018), 'Revisiting small batch training for deep neural networks', *arXiv preprint arXiv:1804.07612* .

Mehran, R., Oyama, A. and Shah, M. (2009), Abnormal crowd behavior detection using social force model, *in* 'Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on', IEEE, pp. 935–942.

Misra, I., Shrivastava, A., Gupta, A. and Hebert, M. (2016), Cross-stitch networks for multi-task learning, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 3994–4003.

Mohammadi, S., Perina, A., Kiani, H. and Murino, V. (2016), Angry crowds: detecting violent events in videos, *in* 'European Conference on Computer Vision', Springer, pp. 3–18.

Ng, J. Y.-H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R. and Toderici, G. (2015), Beyond short snippets: Deep networks for video classification, *in* 'Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on', IEEE, pp. 4694–4702.

Onoro-Rubio, D. and López-Sastre, R. J. (2016), Towards perspective-free object counting with deep learning, *in* 'European Conference on Computer Vision', Springer, pp. 615–629.

Pan, S. J. and Yang, Q. (2010), 'A survey on transfer learning', *IEEE Transactions on knowledge and data engineering* **22**(10), 1345–1359.

Powers, D. M. (1998), Applications and explanations of zipf's law, *in* 'Proceedings of the joint conferences on new methods in language processing and computational natural language learning', Association for Computational Linguistics, pp. 151–160.

Ranjan, R., Patel, V. M. and Chellappa, R. (2017), 'Hyperface: A deep multi-task learning

framework for face detection, landmark localization, pose estimation, and gender recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence* .

Ravanbakhsh, M., Sangineto, E., Nabi, M. and Sebe, N. (2017), 'Training adversarial discriminators for cross-channel abnormal event detection in crowds', *arXiv preprint arXiv:1706.07680* .

Rebuffi, S.-A., Bilen, H. and Vedaldi, A. (2017), 'Learning multiple visual domains with residual adapters', *2017 Conference on Neural Information Processing Systems (NIPS)* .

Reddy, V., Sanderson, C. and Lovell, B. C. (2011), Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture, *in* 'Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on', IEEE, pp. 55–61.

Rosenfeld, A. and Tsotsos, J. K. (2017), 'Incremental learning through deep adaptation', *arXiv preprint arXiv:1705.04228* .

Roshtkhari, M. J. and Levine, M. D. (2013), Online dominant and anomalous behavior detection in videos, *in* 'Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on', IEEE, pp. 2611–2618.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986), 'Learning representations by back-propagating errors', *nature* **323**(6088), 533.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. et al. (2015), 'Imagenet large scale visual recognition challenge', *International Journal of Computer Vision* **115**(3), 211–252.

Sam, D. B., Surya, S. and Babu, R. V. (2017), Switching convolutional neural network for crowd counting, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', Vol. 1, p. 6.

Santurkar, S., Tsipras, D., Ilyas, A. and Madry, A. (2018), 'How does batch normalization help optimization?(no, it is not about internal covariate shift)', *arXiv preprint arXiv:1805.11604* .

Seltzer, M. L. and Droppo, J. (2013), Multi-task learning in deep neural networks for improved phoneme recognition, *in* 'Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on', IEEE, pp. 6965–6969.

Senst, T., Eiselein, V., Kuhn, A. and Sikora, T. (2017), 'Crowd violence detection using global motion-compensated lagrangian features and scale-sensitive video-level representation', *IEEE Transactions on Information Forensics and Security* **12**(12), 2945–2956.

Shah, S. (2018), 'Gartner: Four strategies to make smart cities work for citizens', https://internetofbusiness.com/cities-smart-citizens-gartner/. [Online; accessed 12-June-2018].

Shao, J., Kang, K., Change Loy, C. and Wang, X. (2015), Deeply learned attributes for crowded scene understanding, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 4657–4666.

Shao, J., Loy, C.-C., Kang, K. and Wang, X. (2016), Slicing convolutional neural network for crowd video understanding, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 5620–5628.

Simonyan, K. and Zisserman, A. (2014), 'Very deep convolutional networks for large-scale image recognition', *arXiv preprint arXiv:1409.1556* .

Sindagi, V. A. and Patel, V. M. (2017), Generating high-quality crowd density maps using contextual pyramid cnns, *in* '2017 IEEE International Conference on Computer Vision (ICCV)', IEEE, pp. 1879–1888.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015), Going deeper with convolutions, *in* 'The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)'.

Tran, D., Bourdev, L., Fergus, R., Torresani, L. and Paluri, M. (2015), Learning spatiotemporal features with 3d convolutional networks, *in* 'Computer Vision (ICCV), 2015 IEEE International Conference on', IEEE, pp. 4489–4497.

Wang, S., Zhao, H., Wang, W., Di, H. and Shu, X. (2017), Improving deep crowd density estimation via pre-classification of density, *in* 'International Conference on Neural Information Processing', Springer, pp. 260–269.

Wu, B. and Nevatia, R. (2005), Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors, *in* 'Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1', Vol. 1, IEEE, pp. 90–97.

Xiaohua, L., Lansun, S. and Huanqin, L. (2006), 'Estimation of crowd density based on wavelet and support vector machine', *Transactions of the Institute of Measurement and Control* **28**(3), 299–308.

Xie, W., Noble, J. A. and Zisserman, A. (2016), 'Microscopy cell counting and detection with fully convolutional regression networks', *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* pp. 1–10.

Xu, D., Ricci, E., Yan, Y., Song, J. and Sebe, N. (2015), 'Learning deep representations of appearance and motion for anomalous event detection', *arXiv preprint arXiv:1510.01553* .

Xu, L., Gong, C., Yang, J., Wu, Q. and Yao, L. (2014), Violent video detection based on mosift feature and sparse coding, *in* 'Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on', IEEE, pp. 3538–3542.

Yan, Y., Ricci, E., Subramanian, R., Liu, G., Lanz, O. and Sebe, N. (2016), 'A multi-task learning framework for head pose estimation under target motion', *IEEE transactions on pattern analysis and machine intelligence* **38**(6), 1070–1083.

Zhang, D., Shen, D., Initiative, A. D. N. et al. (2012), 'Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease', *NeuroImage* **59**(2), 895–907.

Zhang, S., Wu, G., Costeira, J. P. and Moura, J. M. (2017), 'Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras', *arXiv preprint arXiv:1707.09476* .

Zhang, S. and Zhang, X. (2015), 'Pedestrian density estimation by a weighted bag of visual words model', *International Journal of Machine Learning and Computing* **5**(3), 214.

Zhang, Y., Zhou, D., Chen, S., Gao, S. and Ma, Y. (2016), Single-image crowd counting via multi-column convolutional neural network, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 589–597.